TEI P5: Guidelines for Electronic Text Encoding and Interchange

by the TEI Consortium

Originally edited by C.M. Sperberg-McQueen and Lou Burnard for the ACH-ALLC-ACL Text Encoding Initiative
Now entirely revised and expanded under the supervision of the Technical Council of the TEI Consortium

edited by Lou Burnard and Syd Bauman

Oxford — Providence — Charlottesville — Nancy
2008

The TEI Guidelines

edited by Lou Burnard and Syd Bauman

2008

Contents

i Releases of the TEI Guidelines **xv**

ii Dedication **xvii**

iii Preface and Acknowledgments **xix**

iv About These Guidelines **xxiii**
iv.1 Structure and Notational Conventions of this Document . . . xxiv
iv.1.1 Design Principles . . . xxv
iv.1.2 Intended Use . . . xxvi
iv.2 Historical Background . . . xxix
iv.3 Future Developments . . . xxx

v A Gentle Introduction to XML **xxxi**
v.1 What's special about XML? . . . xxxii
v.1.1 Descriptive markup . . . xxxii
v.1.2 Types of document . . . xxxii
v.1.3 Data independence . . . xxxiii
v.2 Textual structures . . . xxxiii
v.3 XML structures . . . xxxiv
v.3.1 Elements . . . xxxiv
v.3.2 Content models: an example . . . xxxiv
v.3.3 Validating a document's structure . . . xxxvi
v.3.4 An example schema . . . xxxvii
v.4 Complicating the issue . . . xl
v.5 Attributes . . . xlii
v.5.1 Declaring attributes . . . xlii
v.5.2 Identifiers and indicators . . . xliv
v.6 Other components of an XML document . . . xlv
v.6.1 Character References . . . xlv
v.6.2 Processing instructions . . . xlvi
v.6.3 Namespaces . . . xlvii
v.7 Putting it all together . . . xlix
v.7.1 Associating entity definitions with a document instance . . . l
v.7.2 Associating a document instance with its schema . . . l
v.7.3 Assembling multiple resources into a single document . . . li
v.7.4 Stylesheet association and processing . . . li

vi Languages and Character Sets **liii**
vi.1 Language identification . . . liv
vi.2 Characters and Character Sets . . . lvi
vi.2.1 Historical considerations . . . lvi
vi.2.2 Terminology and key concepts . . . lvii
vi.2.3 Abstract characters, glyphs and encoding scheme design . . . lviii
vi.2.4 Entry of characters. . . . lix
vi.2.5 Output of characters . . . lx
vi.2.6 Unicode and XML . . . lx

vi.2.7 Special aspects of Unicode character definitions . . . lxiii
vi.2.8 Character entities in non-validated documents . . . lxiv
vi.2.9 Issues arising from the internal representations of Unicode . . . lxv

1 The TEI Infrastructure 1
1.1 TEI Modules . . . 2
1.2 Defining a TEI Schema . . . 3
1.2.1 A Simple Customization . . . 3
1.2.2 A Larger Customization . . . 4
1.3 The TEI Class System . . . 5
1.3.1 Attribute Classes . . . 5
1.3.2 Model Classes . . . 9
1.4 Macros . . . 12
1.4.1 Standard Content Models . . . 12
1.4.2 Datatype Macros . . . 12
1.5 The TEI Infrastructure Module . . . 14

2 The TEI Header 17
2.1 Organization of the TEI Header . . . 18
2.1.1 The TEI Header and its Components . . . 18
2.1.2 Types of Content in the TEI Header . . . 19
2.1.3 Model Classes in the TEI Header . . . 20
2.2 The File Description . . . 21
2.2.1 The Title Statement . . . 22
2.2.2 The Edition Statement . . . 24
2.2.3 Type and Extent of File . . . 25
2.2.4 Publication, Distribution, etc. . . . 26
2.2.5 The Series Statement . . . 28
2.2.6 The Notes Statement . . . 28
2.2.7 The Source Description . . . 30
2.2.8 Computer Files Derived from Other Computer Files . . . 32
2.3 The Encoding Description . . . 32
2.3.1 The Project Description . . . 33
2.3.2 The Sampling Declaration . . . 33
2.3.3 The Editorial Practices Declaration . . . 34
2.3.4 The Tagging Declaration . . . 36
2.3.5 The Reference System Declaration . . . 40
2.3.6 The Classification Declaration . . . 42
2.3.7 The Application Information Element . . . 44
2.3.8 Module-Specific Declarations . . . 45
2.4 The Profile Description . . . 45
2.4.1 Creation . . . 46
2.4.2 Language Usage . . . 46
2.4.3 The Text Classification . . . 47
2.5 The Revision Description . . . 49
2.6 Minimal and Recommended Headers . . . 50
2.7 Note for Library Cataloguers . . . 53
2.8 The TEI Header Module . . . 54

3 Elements Available in All TEI Documents **55**
3.1 Paragraphs . . . 56
3.2 Treatment of Punctuation . . . 57
3.3 Highlighting and Quotation . . . 58
3.3.1 What Is Highlighting? . . . 59
3.3.2 Emphasis, Foreign Words, and Unusual Language . . . 60
3.3.3 Quotation . . . 63
3.3.4 Terms, Glosses, Equivalents, and Descriptions . . . 67
3.3.5 Some Further Examples . . . 70
3.4 Simple Editorial Changes . . . 72
3.4.1 Apparent Errors . . . 73
3.4.2 Regularization and Normalization . . . 74
3.4.3 Additions, Deletions, and Omissions . . . 76
3.5 Names, Numbers, Dates, Abbreviations, and Addresses . . . 78
3.5.1 Referring Strings . . . 78
3.5.2 Addresses . . . 81
3.5.3 Numbers and Measures . . . 83
3.5.4 Dates and Times . . . 85
3.5.5 Abbreviations and Their Expansions . . . 87
3.6 Simple Links and Cross-References . . . 88
3.7 Lists . . . 91
3.8 Notes, Annotation, and Indexing . . . 95
3.8.1 Notes and Simple Annotation . . . 95
3.8.2 Index Entries . . . 96
3.9 Graphics and other non-textual components . . . 101
3.10 Reference Systems . . . 102
3.10.1 Using the xml:id and n Attributes . . . 103
3.10.2 Creating New Reference Systems . . . 105
3.10.3 Milestone Elements . . . 106
3.10.4 Declaring Reference Systems . . . 109
3.11 Bibliographic Citations and References . . . 111
3.11.1 Elements of Bibliographic References . . . 112
3.11.2 Components of Bibliographic References . . . 115
3.11.3 Bibliographic Pointers . . . 125
3.11.4 Relationship to Other Bibliographic Schemes . . . 126
3.12 Passages of Verse or Drama . . . 127
3.12.1 Core Tags for Verse . . . 128
3.12.2 Core Tags for Drama . . . 130
3.13 Overview of the Core Module . . . 134

4 Default Text Structure **135**
4.1 Divisions of the Body . . . 137
4.1.1 Un-numbered Divisions . . . 137
4.1.2 Numbered Divisions . . . 138
4.1.3 Numbered or Un-numbered? . . . 139
4.1.4 Partial and Composite Divisions . . . 142
4.2 Elements Common to All Divisions . . . 144
4.2.1 Headings and Trailers . . . 145

4.2.2 Openers and Closers . . . 146
4.2.3 Arguments, Epigraphs, and Postscripts . . . 148
4.2.4 Content of Textual Divisions . . . 150
4.3 Grouped and Floating Texts . . . 150
4.3.1 Grouped Texts . . . 151
4.3.2 Floating Texts . . . 158
4.4 Virtual Divisions . . . 160
4.5 Front Matter . . . 160
4.6 Title Pages . . . 163
4.7 Back Matter . . . 165
4.8 Module for Default Text Structure . . . 167

5 Representation of Non-standard Characters and Glyphs 169
5.1 Is Your Journey Really Necessary? . . . 169
5.2 Markup Constructs for Representation of Characters and Glyphs . . . 170
5.2.1 Character Properties . . . 173
5.3 Annotating Characters . . . 176
5.4 Adding New Characters . . . 179
5.5 How to Use Code Points from the Private Use Area . . . 180
5.6 Module Character and Glyph Documentation . . . 181

6 Verse 183
6.1 Structural Divisions of Verse Texts . . . 183
6.2 Components of the Verse Line . . . 187
6.3 Rhyme and Metrical Analysis . . . 189
6.3.1 Sample Metrical Analyses . . . 190
6.3.2 Segment-Level versus Line-level Tagging . . . 192
6.3.3 Metrical Analysis of Stanzaic Verse . . . 193
6.4 Rhyme . . . 194
6.5 Metrical Notation Declaration . . . 196
6.6 Encoding Procedures for Other Verse Features . . . 198
6.7 Module for Verse . . . 198

7 Performance Texts 199
7.1 Front and Back Matter . . . 199
7.1.1 The Set Element . . . 200
7.1.2 Prologues and Epilogues . . . 201
7.1.3 Records of Performances . . . 203
7.1.4 Cast Lists . . . 204
7.2 The Body of a Performance Text . . . 208
7.2.1 Major Structural Divisions . . . 208
7.2.2 Speeches and Speakers . . . 209
7.2.3 Stage Directions . . . 212
7.2.4 Speech Contents . . . 215
7.2.5 Embedded Structures . . . 217
7.2.6 Simultaneous Action . . . 221
7.3 Other Types of Performance Text . . . 222
7.3.1 Technical Information . . . 224

7.4 Module for Performance Texts . . . 224

8 Transcriptions of Speech 225
8.1 General Considerations and Overview . . . 225
8.2 Documenting the Source of Transcribed Speech . . . 227
8.3 Elements Unique to Spoken Texts . . . 230
8.3.1 Utterances . . . 232
8.3.2 Pausing . . . 234
8.3.3 Vocal, Kinesic, Incident . . . 234
8.3.4 Writing . . . 235
8.3.5 Temporal Information . . . 236
8.3.6 Shifts . . . 236
8.4 Elements Defined Elsewhere . . . 239
8.4.1 Segmentation . . . 239
8.4.2 Synchronization and Overlap . . . 240
8.4.3 Regularization of Word Forms . . . 245
8.4.4 Prosody . . . 245
8.4.5 Speech Management . . . 247
8.4.6 Analytic Coding . . . 248
8.5 Module for Transcribed Speech . . . 249

9 Dictionaries 251
9.1 Dictionary Body and Overall Structure . . . 252
9.2 The Structure of Dictionary Entries . . . 254
9.2.1 Hierarchical Levels . . . 254
9.2.2 Groups and Constituents . . . 256
9.3 Top-level Constituents of Entries . . . 259
9.3.1 Information on Written and Spoken Forms . . . 259
9.3.2 Grammatical Information . . . 263
9.3.3 Sense Information . . . 264
9.3.4 Etymological Information . . . 268
9.3.5 Other Information . . . 269
9.3.6 Related Entries . . . 276
9.4 Headword and Pronunciation References . . . 277
9.5 Typographic and Lexical Information in Dictionary Data . . . 280
9.5.1 Editorial View . . . 281
9.5.2 Lexical View . . . 284
9.5.3 Retaining Both Views . . . 285
9.6 Unstructured Entries . . . 289
9.7 The Dictionary Module . . . 290

10 Manuscript Description 291
10.1 Overview . . . 291
10.2 The Manuscript Description Element . . . 291
10.3 Phrase-level Elements . . . 295
10.3.1 Origination . . . 296
10.3.2 Material . . . 296
10.3.3 Watermarks and Stamps . . . 296

10.3.4 Dimensions . . . 297
10.3.5 References to Locations within a Manuscript . . . 298
10.3.6 Names of Persons, Places, and Organizations . . . 299
10.3.7 Catchwords, Signatures, Secundo Folio . . . 301
10.3.8 Heraldry . . . 302
10.4 The Manuscript Identifier . . . 302
10.5 The Manuscript Heading . . . 306
10.6 Intellectual Content . . . 307
10.6.1 The <msItem> and <msItemStruct> Elements . . . 308
10.6.2 Authors and Titles . . . 310
10.6.3 Rubrics, Incipits, Explicits, and Other Quotations from the Text . . . 311
10.6.4 Filiation . . . 312
10.6.5 Text Classification . . . 312
10.6.6 Languages and Writing Systems . . . 312
10.7 Physical Description . . . 313
10.7.1 Object Description . . . 314
10.7.2 Writing, Decoration, and Other Notations . . . 318
10.7.3 Bindings, Seals, and Additional Material . . . 322
10.7.4 History . . . 324
10.7.5 Additional information . . . 325
10.7.6 Manuscript Parts . . . 329
10.7.7 Module for Manuscription Description . . . 329

11 Representation of Primary Sources 331
11.1 Digital Facsimiles . . . 331
11.2 Scope of Transcriptions . . . 341
11.3 Altered, Corrected, and Erroneous Texts . . . 341
11.3.1 Core elements for Transcriptional Work . . . 342
11.3.2 Abbreviation and Expansion . . . 342
11.3.3 Correction and Conjecture . . . 346
11.3.4 Additions and Deletions . . . 351
11.3.5 Substitutions . . . 355
11.3.6 Cancellation of Deletions and Other Markings . . . 357
11.3.7 Text Omitted from or Supplied in the Transcription . . . 357
11.4 Hands and Responsibility . . . 359
11.4.1 Document Hands . . . 359
11.4.2 Hand, Responsibility, and Certainty Attributes . . . 360
11.5 Damage and Conjecture . . . 362
11.5.1 Damage, Illegibility, and Supplied Text . . . 362
11.5.2 Use of the <gap>, <del>, <damage>, <unclear>, and <supplied> Elements in Combination . . . 365
11.6 Aspects of Layout . . . 366
11.6.1 Space . . . 366
11.6.2 Lines . . . 367
11.7 Headers, Footers, and Similar Matter . . . 368
11.8 Other Primary Source Features not Covered in these Guidelines . . . 369
11.9 Module for Transcription of Primary Sources . . . 369

12 Critical Apparatus **371**
12.1 The Apparatus Entry, Readings, and Witnesses . . . 371
12.1.1 The Apparatus Entry . . . 372
12.1.2 Readings . . . 372
12.1.3 Indicating Subvariation in Apparatus Entries . . . 375
12.1.4 Witness Information . . . 378
12.1.5 Fragmentary Witnesses . . . 382
12.2 Linking the Apparatus to the Text . . . 383
12.2.1 The Location-referenced Method . . . 384
12.2.2 The Double End-Point Attachment Method . . . 385
12.2.3 The Parallel Segmentation Method . . . 387
12.3 Using Apparatus Elements in Transcriptions . . . 389
12.4 Module for Critical Apparatus . . . 390

13 Names, Dates, People, and Places **391**
13.1 Attribute Classes Defined by this Module . . . 391
13.1.1 Linking Names and their Referents . . . 391
13.1.2 Dating Attributes . . . 393
13.2 Names . . . 394
13.2.1 Personal Names . . . 394
13.2.2 Organizational Names . . . 400
13.2.3 Place Names . . . 402
13.3 Biographical and Prosopographical Data . . . 405
13.3.1 Basic Principles . . . 406
13.3.2 The Person Element . . . 406
13.3.3 Organizational Data . . . 415
13.3.4 Places . . . 417
13.3.5 Names and Nyms . . . 426
13.3.6 Dates and Times . . . 429
13.4 Module for Names and Dates . . . 433

14 Tables, Formulæ, and Graphics **435**
14.1 Tables . . . 435
14.1.1 TEI Tables . . . 436
14.1.2 Other Table Schemas . . . 439
14.2 Formulæ and Mathematical Expressions . . . 440
14.3 Specific Elements for Graphic Images . . . 443
14.4 Overview of Basic Graphics Concepts . . . 447
14.5 Graphic Image Formats . . . 448
14.5.1 Vector Graphic Formats . . . 448
14.5.2 Raster Graphic Formats . . . 448
14.5.3 Photographic and Motion Video Formats . . . 449
14.6 Module for Tables, Formulæ, and Graphics . . . 450

15 Language Corpora **451**
15.1 Varieties of Composite Text . . . 452
15.2 Contextual Information . . . 454
15.2.1 The Text Description . . . 455

15.2.2 The Participant Description . . . 457
15.2.3 The Setting Description . . . 458
15.3 Associating Contextual Information with a Text . . . 460
15.3.1 Combining Corpus and Text Headers . . . 460
15.3.2 Declarable Elements . . . 461
15.3.3 Summary . . . 464
15.4 Linguistic Annotation of Corpora . . . 465
15.4.1 Levels of Analysis . . . 465
15.5 Recommendations for the Encoding of Large Corpora . . . 466
15.6 Module for Language Corpora . . . 466

16 Linking, Segmentation, and Alignment 467
16.1 Links . . . 468
16.1.1 Pointers and Links . . . 468
16.1.2 Using Pointers and Links . . . 469
16.1.3 Groups of Links . . . 473
16.1.4 Intermediate Pointers . . . 474
16.2 Pointing Mechanisms . . . 475
16.2.1 Pointing Elsewhere . . . 475
16.2.2 Pointing Locally . . . 477
16.2.3 W3C element() Scheme . . . 478
16.2.4 TEI XPointer Schemes . . . 479
16.2.5 Canonical References . . . 482
16.3 Blocks, Segments, and Anchors . . . 485
16.4 Correspondence and Alignment . . . 490
16.4.1 Correspondence . . . 490
16.4.2 Alignment of Parallel Texts . . . 492
16.4.3 A Three-way Alignment . . . 494
16.5 Synchronization . . . 498
16.5.1 Aligning Synchronous Events . . . 498
16.5.2 Placing Synchronous Events in Time . . . 501
16.6 Identical Elements and Virtual Copies . . . 502
16.7 Aggregation . . . 504
16.8 Alternation . . . 509
16.9 Stand-off Markup . . . 514
16.9.1 Introduction . . . 514
16.9.2 Overview of XInclude . . . 515
16.9.3 Doing Stand-off Markup in TEI . . . 516
16.9.4 Well-formedness and Validity of Stand-off Markup . . . 518
16.9.5 Including Text or XML Fragments . . . 519
16.10 Connecting Analytic and Textual Markup . . . 520
16.11 Module for Linking, Segmentation, and Alignment . . . 520

17 Simple Analytic Mechanisms 521
17.1 Linguistic Segment Categories . . . 521
17.2 Global Attributes for Simple Analyses . . . 527
17.3 Spans and Interpretations . . . 527
17.4 Linguistic Annotation . . . 531

17.5 Module for Analysis and Interpretation . . . 535

18 Feature Structures **537**
18.1 Organization of this Chapter . . . 537
18.2 Elementary Feature Structures and the Binary Feature Value . . . 537
18.3 Other Atomic Feature Values . . . 539
18.4 Feature and Feature-Value Libraries . . . 542
18.5 Feature Structures as Complex Feature Values . . . 544
18.6 Re-entrant Feature Structures . . . 546
18.7 Collections as Complex Feature Values . . . 547
18.8 Feature Value Expressions . . . 549
18.8.1 Alternation . . . 549
18.8.2 Negation . . . 552
18.8.3 Collection of Values . . . 553
18.9 Default Values . . . 553
18.10 Linking Text and Analysis . . . 554
18.11 Feature System Declaration . . . 557
18.11.1 Linking a TEI Text to Feature System Declarations . . . 558
18.11.2 The Overall Structure of a Feature System Declaration . . . 560
18.11.3 Feature Declarations . . . 561
18.11.4 Feature Structure Constraints . . . 565
18.11.5 A Complete Example . . . 567
18.12 Formal Definition and Implementation . . . 571

19 Graphs, Networks, and Trees **573**
19.1 Graphs and Digraphs . . . 573
19.1.1 Transition Networks . . . 578
19.1.2 Family Trees . . . 581
19.1.3 Historical Interpretation . . . 584
19.2 Trees . . . 587
19.3 Another Tree Notation . . . 592
19.4 Representing Textual Transmission . . . 600
19.5 Module for Graphs, Networks, and Trees . . . 602

20 Non-hierarchical Structures **605**
20.1 Multiple Encodings of the Same Information . . . 606
20.2 Boundary Marking with Empty Elements . . . 607
20.3 Fragmentation and Reconstitution of Virtual Elements . . . 611
20.4 Stand-off Markup . . . 615
20.5 Non-XML-based Approaches . . . 616

21 Certainty and Responsibility **619**
21.1 Levels of Certainty . . . 619
21.1.1 Using Notes to Record Uncertainty . . . 620
21.1.2 Structured Indications of Uncertainty . . . 620
21.2 Attribution of Responsibility . . . 625
21.3 The Certainty Module . . . 625

22 Documentation Elements **627**
22.1 Phrase Level Documentary Elements . . . 628
22.1.1 Phrase Level Terms . . . 628
22.1.2 Element and Attribute Descriptions . . . 629
22.2 Modules and Schemas . . . 630
22.3 Specification Elements . . . 632
22.4 Common Elements . . . 633
22.4.1 Description of Components . . . 633
22.4.2 Exemplification of Components . . . 634
22.4.3 Classification of Components . . . 635
22.4.4 Element Specifications . . . 635
22.4.5 Attribute List Specification . . . 636
22.4.6 Element Classes . . . 639
22.4.7 Pattern Documentation . . . 642
22.5 Building a Schema . . . 642
22.6 Combining TEI and Non-TEI Modules . . . 644
22.7 Module for Documention Elements . . . 645

23 Using the TEI **647**
23.1 Obtaining the TEI Schemas . . . 647
23.2 Personalization and Customization . . . 647
23.2.1 Kinds of Modification . . . 649
23.2.2 Modification and Namespaces . . . 656
23.2.3 Documenting the Modification . . . 657
23.2.4 Examples of Modification . . . 658
23.3 Conformance . . . 659
23.3.1 Well-formedness criterion . . . 659
23.3.2 Validation Constraint . . . 660
23.3.3 Conformance to the TEI Abstract Model . . . 660
23.3.4 Use of the TEI Namespace . . . 661
23.3.5 Documentation Constraint . . . 662
23.3.6 Varieties of TEI Conformance . . . 662
23.4 Implementation of an ODD System . . . 664
23.4.1 Making a Unified ODD . . . 664
23.4.2 Generating Schemas . . . 668
23.4.3 Names and Documentation in Generated Schemas . . . 672
23.4.4 Making a RELAX NG Schema . . . 673
23.4.5 Making a DTD . . . 678
23.4.6 Generating Documentation . . . 679
23.4.7 Using TEI Parameterized Schema Fragments . . . 679

A Model Classes **685**

B Attribute Classes **705**

C Elements **737**

D Attributes **1227**

E Datatypes and Other Macros **1233**

F Bibliography **1253**
Works cited in examples in the Guidelines . . . 1253
Works cited elsewhere in the text of the Guidelines . . . 1262
Reading list . . . 1266
Theory of Markup and XML . . . 1266
TEI . . . 1272

G Prefatory Notes **1277**
Prefatory Note (March 2002) . . . 1277
Introductory Note (November 2001) . . . 1278
Introductory Note (June 2001) . . . 1278
Introductory Note (May 1999) . . . 1280
Typographic corrections made . . . 1280
Specific changes in the DTD . . . 1280
Outstanding errors . . . 1281
Preface (April 1994) . . . 1282
Acknowledgments . . . 1283
TEI Working Committees (1990-1993) . . . 1283
Advisory Board . . . 1285
Steering Committee Membership . . . 1286

H Colophon **1287**

i

Releases of the TEI Guidelines

P1 1990, C.M. Sperberg-McQueen and Lou Burnard

P2 1992, C.M. Sperberg-McQueen and Lou Burnard

P3 1994, C.M. Sperberg-McQueen and Lou Burnard

P4 2001, Lou Burnard, Syd Bauman, and Steven DeRose

P5 2007, Lou Burnard and Syd Bauman

ii

Dedication

In memoriam
Donald E. Walker
22 November 1928 – 26 November 1993
Antonio Zampolli
1937 – 22 August 2003

iii

Preface and Acknowledgments

This publication constitutes the fifth distinct version of the *Guidelines for Electronic Text Encoding and Interchange*, and the first complete revision since the appearance of P3 in 1994. It includes substantial amounts of new material and a major revision of the underlying technical infrastructure. With this version, the Guidelines enter a new stage in their development as a community-maintained open source project. This edition is the first version to have benefitted from the close overview and oversight of an elected TEI Technical Council. The editors are therefore particularly pleased to acknowledge with gratitude the hard work and dedication put into this project by the Council over the last five years.

The Chair of the TEI Board sits on the Technical Council, and the Board also nominates one other member to the Council. The other Council members are all elected by the Consortium membership, and serve periods of up to two years at a time. The Board nominates the Chair of the Technical Council from among its members. The names and affiliations of all Council members who served during the production of this edition of the Guidelines are listed below.

Chair

- 2002-3: John Unsworth (University of Virginia)
- 2003-7: Christian Wittern (Kyoto University)

Board Members

- 2002-7: Sebastian Rahtz (University of Oxford)
- 2004-5: Julia Flanders (Brown University)
- 2006: Matthew Zimmerman (New York University)
- 2007: Daniel O'Donnell (University of Lethbridge)

Elected Members

- 2003-6: Alejandro Bia (University of Alicante)
- 2004-6; 2006-7: David Birnbaum (University of Pittsburgh)
- 2007: Tone Merete Bruvik (University of Bergen)
- 2007: Arianna Ciula (King's College London)
- 2005-7: James Cummings (University of Oxford)
- 2002-7: Matthew Driscoll (University of Copenhagen)
- 2002-4: David Durand (Ingenta plc)

- 2002-4: Tomas Erjavec (Jozef Stefan Institute, Ljubljana)
- 2002: Fotis Jannidis (University of Munich)
- 2006: Amit Kumar (University of Illinois at Urbana-Champaign)
- 2002: Martin Mueller (Northwestern University)
- 2006-7: Dorothy Porter (University of Kentucky)
- 2002-3: Merillee Proffitt (Research Libraries Group)
- 2002: Peter Robinson (De Montfort University)
- 2002: Geoffrey Rockwell (Macmaster University)
- 2002-7: Laurent Romary (University of Nancy; Max Planck Digital Library)
- 2003-7: Susan Schreibman (University of Maryland)
- 2004-5: Natasha Smith (University of North Carolina at Chapel Hill)
- 2006-7: Conal Tuohy (Victoria University of Wellington)
- 2004-5: Edward Vanhoutte (Royal Academy of Dutch Language and Literature)
- 2005-7: John Walsh (Indiana University)
- 2002-5: Perry Willett (Indiana University)

The bulk of the Council's work has been carried out by email and by regular telephone conference. In addition, the Council has held six two-day face-to-face meetings. During production of P5, these meetings were generously hosted by the following institutions:

King's College, London (2002)

Oxford University Computing Services (2003)

Royal Academy of Dutch Language and Literature, Ghent (2004)

AFNOR: Association française de normalisation, Paris (2005)

Institute for Research in Humanities, Kyoto University (2006)

Berlin-Brandenburgische Akademie der Wissenschaften, Berlin (2007)

During the production of TEI P5, the Council chartered a number of smaller workgroups and similar activities, each of which made significant contribution to the intellectual content of the work. Active members of these are listed below:

Character Set Workgroup Active between July 2001 and January 2005, this group revised and developed the recommendations now forming chapters *vi Languages and Character Sets* and *5. Representation of Non-standard Characters and Glyphs.* It was chaired by Christian Wittern, and its membership included: Deborah Anderson (Berkeley); Michael Beddow (independent scholar); David Birnbaum (Pittsburgh University); Martin Duerst (W3C/Keio University); Patrick Durusau (Society of Biblical Literature); Tomohiko Morioka (Kyoto University); and Espen Ore (National Library of Norway).

Meta Taskforce Active between February 2003 and February 2005, this group developed the material now forming *22. Documentation Elements.* It was chaired by Sebastian Rahtz, and its membership included: Alejandro Bia; David G. Durand; Laurent Romary; Norman Walsh (Sun Microsystems); and Christian Wittern.

Workgroup on Stand-Off Markup, XLink and XPointer Active between February 2002 and January 2006, this group reviewed and expanded the material now largely forming part of *16. Linking, Segmentation, and Alignment*. It was chaired by David G. Durand, and its membership included: Jean Carletta (Edinburgh University); Chris Caton (University of Oxford); Jessica P. Hekman (Ingenta plc); Nancy M. Ide (Vassar College); and Fabio Vitali (University of Bologna).

Manuscript Description Task Force Active between February 2003 and December 2005, this group reviewed and finalised the material now forming *10. Manuscript Description*. It was chaired by Matthew Driscoll and comprised David Birnbaum and Merrillee Proffitt, in addition to the TEI Editors.

Names and Places Activity Active between January 2006 and May 2007, this group formulated the new material now forming part of *13. Names, Dates, People, and Places*. It was chaired by Matthew Driscoll. and its membership included Gabriel Bodard (King's College London); Arianna Ciula; James Cummings; Tom Elliott (University of North Carolina at Chapel Hill); Øyvind Eide (University of Oslo); Leif Isaksen (Oxford Archaeology plc); Richard Light (private consultant); Tadeusz Piotrowski (Opole University); Sebastian Rahtz; and Tatiana Timcenko (Vilnius University).

Joint TEI/ISO Activity on Feature Structures Active between January 2003 and August 2007, this group reviewed the material now presented in *18. Feature Structures* and revised it for inclusion in ISO Standard 24610. It was chaired by Kiyong Lee (Korea University), and its active membership included the following: Harry Bunt (Tilburg); Lionel Clément (INRIA); Eric de la Clergerie (INRIA); Thierry Declerck (Saarbrücken); Patrick Drouin (University of Montréal); Lee Gillam (Surrey University); and Kōiti Hasida (ICOT).

The TEI Editors, Lou Burnard (University of Oxford) and Syd Bauman (Brown University) serve *ex officio* on the Council and, as far as possible, on all Council workgroups.

The council also oversees an Internationalization and Localization project, led by Sebastian Rahtz and with funding from the ALLC. This activity, ongoing since October 2005, is engaged in translating key parts of the P5 source into a variety of languages.

Production of the translations currently included in P5 has been co-ordinated by the following:

Chinese Marcus Bingenheimer (Chung-hwa Institute of Buddhist Studies, Taipei) and Weining Hwang (Würzburg University)

French Pierre-Yves Duchemin (ENSSIB); Jean-Luc Benoit (ATILF); Anila Angjeli (BnF); Joëlle Bellec Martini (BnF); Marie-France Claerebout (Aldine); Magali Le Coënt (BIUSJ); Florence Clavaud (EnC); Cécile Pierre (BIUSJ).

German Werner Wegstein (Würzburg University)

Japanese Ohya Kazushi (Tsurumi University)

Spanish Carmen Arronis Llopis (University of Alicante) and Alejandro Bia (Miguel Hernández University)

Italian Marco Venuti (University of Venice) and Letizia Cirillo (University of Bologna)

iv

About These Guidelines

These Guidelines have been developed and are maintained by the Text Encoding Initiative Consortium (TEI); see *iv.2 Historical Background*. They are addressed to anyone who works with any kind of textual resource in digital form.

They make recommendations about suitable ways of representing those features of textual resources which need to be identified explicitly in order to facilitate processing by computer programs. In particular, they specify a set of markers (or tags) which may be inserted in the electronic representation of the text, in order to mark the text structure and other features of interest. Many, or most, computer programs depend on the presence of such explicit markers for their functionality, since without them a digitized text appears to be nothing but a sequence of undifferentiated bits. The success of the World Wide Web, for example, is partly a consequence of its use of such markup to indicate such features as headings and lists on individual pages, and to indicate links between pages. The process of inserting such explicit markers for implicit textual features is often called 'markup', or equivalently within this work 'encoding'; the term 'tagging' is also used informally. We use the term encoding scheme or markup language to denote the complete set of rules associated with the use of markup in a given context; we use the term markup vocabulary for the specific set of markers or named distinctions employed by a given encoding scheme. Thus, this work both describes the TEI encoding scheme, and documents the TEI markup vocabulary.

The TEI encoding scheme is of particular usefulness in facilitating the loss-free interchange of data amongst individuals and research groups using different programs, computer systems, or application software. Since they contain an inventory of the features most often deployed for computer-based text processing, the Guidelines are also useful as a starting point for those designing new systems and creating new materials, even where interchange of information is not a primary objective.

These Guidelines apply to texts in any natural language, of any date, in any literary genre or text type, without restriction on form or content. They treat both continuous materials ('running text') and discontinuous materials such as dictionaries and linguistic corpora. Though principally directed to the needs of the scholarly research community, the Guidelines are not restricted to esoteric academic applications. They are also useful for librarians maintaining and documenting electronic materials, and for publishers and others creating or distributing electronic texts. Although they focus on problems of representing in electronic form texts which already exist in traditional media, these Guidelines are also applicable to textual material which is 'born digital'. We believe them to be adequate to the widest variety of currently existing practices in using digital textual data, but by no means limited to them.

The rules and recommendations made in these Guidelines are expressed in terms of what is currently the most widely-used markup language for digital resources of all kinds: the Extensible Markup Language (XML), as defined by the World Wide Web Consortium's XML Recommendation. However, the TEI encoding scheme itself does not depend on this language; it was originally formulated in terms of a predecessor of XML (the ISO

Standard Generalized Markup Language), and may in future years be re-expressed in other such frameworks as the field of markup develops and matures. For more information on markup languages see chapter *v A Gentle Introduction to XML*; for more information on the associated character encoding issues see chapter *vi Languages and Character Sets*.

This document provides the authoritative and complete statement of the requirements and usage of the TEI encoding scheme. As such, although it includes numerous small examples, it must be stressed that this work is intended to be a reference manual rather than a tutorial guide.

The remainder of this chapter comprises three sections. The first gives an overview of the structure and notational conventions used throughout these Guidelines. The second enumerates the design principles underlying the TEI scheme and the application environments in which it may be found useful. Finally, the third section gives a brief account of the origins and development of the Text Encoding Initiative itself.

iv.1 Structure and Notational Conventions of this Document

The remaining two sections of the front matter to the Guidelines provide background tutorial material for those unfamiliar with basic markup technologies. Following the present introductory section, we present a detailed introduction to XML itself, intended to cover in a relatively painless manner as much as the novice user of the TEI scheme needs to know about markup languages in general and XML in particular. This is followed by a discussion of the general principles underlying current practice in the representation of different languages and writing systems in digital form. This chapter is largely intended for the user unfamiliar with the Unicode encoding systems, though the expert may also find its historical overview of interest.

The body of this edition of the Guidelines proper contains 23 chapters arranged in increasing order of specialist interest. The first five chapters discuss in depth matters likely to be of importance to anyone intending to apply the TEI scheme to virtually any kind of text. The next seven focus on particular kinds of text: verse, drama, spoken text, dictionaries, and manuscript materials. The next nine chapters deal with a wide range of topics, one or more of which are likely to be of interest in specialist applications of various kinds. The last two chapters deal with the XML encoding used to represent the TEI scheme itself, and provide technical information about its implementation. The last chapter also defines the notion of TEI conformance and its implications for interchange of materials produced according to these Guidelines.

As noted above, this is a reference work, and is not intended to be read through from beginning to end. However, the reader wishing to understand the full potential of the TEI scheme will need a thorough grasp of the material covered by the first four chapters and the last two. Beyond that, the reader is recommended to select according to their specific interests: one of the strengths of the TEI architecture is its modular nature.

As far as possible, extensive cross referencing is provided wherever related topics are dealt with; these are particularly effective in the online version of the Guidelines. In addition, a series of technical appendixes provide detailed formal definitions for every element, every class, and every macro discussed in the body of the work; these are also cross linked as appropriate. Finally, a detailed bibliography is provided, which identifies the source of many examples cited in the text as well as documenting works referred to, and listing other relevant publications.

As an aid to the reader, most chapters of these Guidelines follow the same basic organization. The chapter begins with an overview of the subjects treated within it, linked to the following subsections. Within each section where new elements are described, a summary table is first given, which provides their names and a brief description of their intended usage. This is then followed where appropriate by further discussion of each element, including wherever possible usage examples taken somewhat eclectically from a variety of real sources. These examples are not intended to be exhaustive, but rather to suggest typical ways in which the elements concerned may usefully be applied. Where appropriate, a link to a statement of the source for most examples is provided in the online version. Within the examples, use of whitespace such as newlines or indentation is simply intended to aid legibility, and is not prescriptive or normative.

Wherever TEI elements or classes are mentioned in the text, they are linked in the online version to the relevant reference specification for the element or class concerned. Element names are always given in the form <name>, where 'name' is the generic identifier of the element; empty elements such as <pb> or <anchor> include a closing slash to distinguish them wherever they are discussed. References to attributes take the form **attname**, where 'attname' is the name of the attribute. References to classes are also presented as links, for example model.divLike for a model class, and att.global for an attribute class.

iv.1.1 Design Principles

Because of its roots in the humanities research community, the TEI scheme is driven by its original goal of serving the needs of research, and is therefore committed to providing a maximum of comprehensibility, flexibility, and extensibility. More specific design goals of the TEI have been that the Guidelines should:

- provide a standard format for data interchange
- provide guidance for the encoding of texts in this format
- support the encoding of all kinds of features of all kinds of texts studied by researchers
- be application independent

This has led to a number of important design decisions, such as:

- the choice of XML and Unicode
- the provision of a large predefined tag set
- encodings for different views of text
- alternative encodings for the same textual features
- mechanisms for user-defined modification of the scheme

We discuss some of these goals in more detail below.

The goal of creating a common interchange format which is application independent requires the definition of a specific markup syntax as well as the definition of a large set of elements or concepts. The syntax of the recommendations made in this document conforms to the World Wide Web Consortium's XML Recommendation (Bray et al. (eds.) (2006)) but their definition is as far as possible independent of any particular schema language.

The goal of providing guidance for text encoding suggests that recommendations be made as to what textual features should be recorded in various situations. However, when selecting certain features for encoding in preference to others, these Guidelines have tended to prefer generic solutions to specific ones, and to avoid areas where no consensus exists, while attempting to accommodate as many diverse views as feasible. Consequently, the TEI Guidelines make (with relatively rare exceptions) no suggestions or restrictions as to the relative importance of textual features. The philosophy of the Guidelines is 'if you want to encode this feature, do it this way' — but very few features are mandatory. In the same spirit, while the Guidelines very rarely require you to encode any particular feature, they do require you to be honest about which features you have encoded, that is, to respect the meanings and usage rules they recommend for specific elements and attributes proposed.

The requirement to support all kinds of materials likely to be of interest in research has largely conditioned the development of the TEI into a very flexible and modular system. The development of other XML vocabularies or standards is typically motivated by the desire to create a single fully specified encoding scheme for use in a well-defined application domain. By contrast, the TEI is intended for use in a large number of rather ill-defined and often overlapping domains. It achieves its generality by means of the modular architecture described in *1. The TEI Infrastructure* which enables each user to create a schema appropriate to their needs without compromising the interoperability of their data.

The Guidelines have been written largely with a focus on text capture (i.e. the representation in electronic form of an already existing copy text in another medium) rather than text creation (where no such copy text

exists). Hence the frequent use of terms like 'transcription', 'original', 'copy text', etc. However, the Guidelines are equally applicable to text creation.

Concerning text capture the TEI Guidelines do not specify a particular approach to the problem of fidelity to the source text and recoverability of the original; such a choice is the responsibility of the text encoder. The current version of these Guidelines, however, provides a more fully elaborated set of tags for markup of rhetorical, linguistic, and simple typographic characteristics of the text than for detailed markup of page layout or for fine distinctions among type fonts or manuscript hands. It should be noted also that, with the present version of the Guidelines, it is no longer necessarily the case that an unmediated version of the source text can be recovered from an encoded text simply by removing the markup.

In these Guidelines, no hard and fast distinction is drawn between 'objective' and 'subjective' information or between 'representation' and 'interpretation'. These distinctions, though widely made and often useful in narrow, well-defined contexts, are perhaps best interpreted as distinctions between issues on which there is a scholarly consensus and issues where no such consensus exists. Such consensus has been, and no doubt will be, subject to change. The TEI Guidelines do not make suggestions or restrictions as to which of these features should be encoded. The use of the terms descriptive and interpretive about different types of encoding in the Guidelines is not intended to support any particular view on these theoretical issues. Historically, it reflects a purely practical division of responsibility amongst the original working committees (see further *iv.2 Historical Background*).

In general, the accuracy and the reliability of the encoding and the appropriateness of the interpretation is for the individual user of the text to determine. The Guidelines provide a means of documenting the encoding in such a way that a user of the text can know the reasoning behind that encoding, and the general interpretive decisions on which it is based. The TEI header may be used to document and justify many such aspects of the encoding, but the choice of TEI elements for a particular feature is in itself a statement about the interpretation reached by the encoder.

In many situations more than one view of a text is needed since no absolute recommendation to embody one specific view of text can apply to all texts and all approaches to them. Within limits, the syntax of XML ensures that some encodings can be ignored for some purposes. To enable encoding multiple views, these Guidelines not only treat a variety of textual features, but sometimes provide several alternative encodings for what appear to be identical textual phenomena. These Guidelines offer the possibility of encoding many different views of the text, simultaneously if necessary. Where different views of the formal structure of a text are required, as opposed to different annotations on a single structural view, however, the formal syntax of XML (which requires a single hierarchical view of text structure) poses some problems; recommendations concerning ways of overcoming or circumventing that restriction are discussed in chapter *20. Non-hierarchical Structures*.

In brief, the TEI Guidelines define a general-purpose encoding scheme which makes it possible to encode different views of text, possibly intended for different applications, serving the majority of scholarly purposes of text studies in the humanities. Because no predefined encoding scheme can possibly serve all research purposes, the TEI scheme is designed to facilitate both selection from a wide range of predefined markup choices, and the addition of new (non-TEI) markup options. By providing a formally verifiable means of extending the TEI recommendations, the TEI makes it simple for such user-identified modifications to be incorporated into future releases of the Guidelines as they evolve. The underlying mechanisms which support these aspects of the scheme are introduced in chapter *1. The TEI Infrastructure*, and detailed discussions of their use provided in chapter *23. Using the TEI*.

iv.1.2 Intended Use

We envisage three primary functions for these Guidelines:

- guidance for individual or local practice in text creation and data capture;

- support of data interchange;
- support of application-independent local processing.

These three functions are so thoroughly interwoven in practice that it is hardly possible to address any one without addressing the others. However, the distinction provides a useful framework for discussing the possible role of the Guidelines in work with electronic texts.

Use in Text Capture and Text Creation

The description of textual features found in the chapters which follow should provide a useful checklist from which scholars planning to create electronic texts should select the subset of features suitable for their project.

Problems specific to text creation or text 'capture' have not been considered explicitly in this document. These Guidelines are not concerned with the process by which a digital text comes into being: it can be typed by hand, scanned from a printed book or typescript, read from a typesetter's tape, or acquired from another researcher who may have used another markup scheme (or no explicit markup at all).

We include here only some general points which are often raised about markup and the process of data capture.

XML can appear distressingly verbose, particularly when (as in these Guidelines) the names of tags and attributes are chosen for clarity and not for brevity. Editor macros and keyboard shortcuts can allow a typist to enter frequently used tags with single keystrokes. It is often possible to transform word-processed or scanned text automatically. Markup-aware software can help with maintaining the hierarchical structure of the document, and display the document with visual formatting rather than raw tags.

The techniques described in chapter *23.2. Personalization and Customization* may be used to develop simpler data capture TEI-conformant schemas, for example with limited numbers of elements, or with shorter names for the tags being used most often. Documents created with such schemas may then be automatically converted to a more elaborated TEI form.

Use for Interchange

The TEI format may simply be used as an interchange format, permitting projects to share resources even when their local encoding schemes differ. If there are **n** different encoding formats, to provide mappings between each possible pair of formats requires n*(n-1) translations; with an interchange format, only 2n such mappings are needed. However, for such translations to be carried out without loss of information, the interchange format chosen must be as expressive (in a formal sense) as any of the target formats; this is a further reason for the TEI's provision of both highly abstract or generic encodings and highly specific ones.

To translate between any pair of encoding schemes implies:

1. identifying the sets of textual features distinguished by the two schemes;
2. determining where the two sets of features correspond;
3. creating a suitable set of mappings.

For example, to translate from encoding scheme X into the TEI scheme:

1. Make a list of all the textual features distinguished in X.
2. Identify the corresponding feature in the TEI scheme. There are three possibilities for each feature:
   (a) the feature exists in both X and the TEI scheme;
   (b) X has a feature which is absent from the TEI scheme;
   (c) X has a feature which corresponds with more than one feature in the TEI scheme.

The first case is a trivial renaming. The second will require an extension to the TEI scheme, as described in chapter *23.2. Personalization and Customization*. The third is more problematic, but not impossible, provided that a consistent choice can be made (and documented) amongst the alternatives.

The ease with which this translation can be defined will of course depend on the clarity with which scheme X represents the features it encodes.

Translating from the TEI into scheme X follows the same pattern, except that if a TEI feature has no equivalent in X, and X cannot be extended, information must be lost in translation.

The rules defining conformance to the Guidelines are given in some detail in chapter *23.3. Conformance*. The basic principles informing those rules may be summarized as follows:

1. The TEI abstract model (that is, the set of categorical distinctions which it defines) must be respected. The correspondence between a tag X and the semantic function assigned to it by these Guidelines may not be changed; such changes are known as tag abuse and strongly deprecated.

2. A TEI document must be expressed as a valid XML-conformant document which uses the TEI namespace appropriately. If, for example, the document encodes features not provided by the Guidelines, such extensions may not be associated with the TEI namespace.

3. It must be possible to validate a TEI document against a schema derived from these Guidelines, possibly with extensions provided in the recommended manner.

Use for Local Processing

Machine-readable text can be manipulated in many ways; some users:

- edit texts (e.g. word processors, syntax-directed editors)
- edit, display, and link texts in hypertext systems
- format and print texts using desktop publishing systems, or batch-oriented formatting programs
- load texts into free-text retrieval databases or conventional databases
- unload texts from databases as search results or for export to other software
- search texts for words or phrases
- perform content analysis on texts
- collate texts for critical editions
- scan texts for automatic indexing or similar purposes
- parse texts linguistically
- analyze texts stylistically
- scan verse texts metrically
- link text and images

These applications cover a wide range of likely uses but are by no means exhaustive. The aim has been to make the TEI Guidelines useful for encoding the same texts for different purposes. We have avoided anything which would restrict the use of the text for other applications. We have also tried not to omit anything essential to any single application.

Because the TEI format is expressed using XML, almost any modern text processing system is able to process it, and new TEI-aware software systems are able to build on a solid base of existing software libraries.

iv.2 Historical Background

The Text Encoding Initiative grew out of a planning conference sponsored by the Association for Computers and the Humanities (ACH) and funded by the U.S. National Endowment for the Humanities (NEH), which was held at Vassar College in November 1987. At this conference some thirty representatives of text archives, scholarly societies, and research projects met to discuss the feasibility of a standard encoding scheme and to make recommendations for its scope, structure, content, and drafting. During the conference, the Association for Computational Linguistics and the Association for Literary and Linguistic Computing agreed to join ACH as sponsors of a project to develop the Guidelines. The outcome of the conference was a set of principles (the 'Poughkeepsie Principles', Burnard (1988)), which determined the further course of the project.

The Text Encoding Initiative project began in June 1988 with funding from the NEH, soon followed by further funding from the Commission of the European Communities, the Andrew W. Mellon Foundation, and the Social Science and Humanities Research Council of Canada. Four working committees, composed of distinguished scholars and researchers from both Europe and North America, were named to deal with problems of text documentation, text representation, text analysis and interpretation, and metalanguage and syntax issues. Each committee was charged with the task of identifying 'significant particularities' in a range of texts, and two editors appointed to harmonise the resulting recommendations.

A first draft version (P1, with the 'P' here and subsequently standing for 'Proposal') of the Guidelines was distributed in July 1990 under the title *Guidelines for the Encoding and Interchange of Machine-Readable Texts*. Extensive public comment and further work on areas not covered in this version resulted in the drafting of a revised version, TEI P2, distribution of which began in April 1992. This version included substantial amounts of new material, resulting from work carried out by several specialist working groups, set up in 1990 and 1991 to propose extensions and revisions to the text of P1. The overall organization, both of the draft itself and of the scheme it describes, was entirely revised and reorganized in response to public comment on the first draft.

In June 1993 an Advisory Board met to review the current state of the TEI Guidelines, and recommended the formal publication of the work done to that time. That version of the TEI Guidelines, TEI P3, consolidated the work published as parts of TEI P2, along with some additional new material and was finally published in May of 1994 without the label *draft*, thus marking the conclusion of the initial development work.

In February of 1998 the World Wide Web Consortium issued a final Recommendation for the Extensible Markup Language, XML.[1] Following the rapid take-up of this new standard metalanguage, it became evident that the TEI Guidelines (which had been published originally as an SGML application) needed to be re-expressed in this new formalism if they were to survive. The TEI editors, with abundant assistance from others who had developed and used TEI, developed an update plan, and made tentative decisions on relevant syntactic issues.

In January of 1999, the University of Virginia and the University of Bergen formally proposed the creation of an international membership organization, to be known as the TEI Consortium, which would maintain, develop, and promote the TEI. Shortly thereafter, two further institutions with longstanding ties to the TEI (Brown University and Oxford University) joined them in formulating an Agreement to Establish a Consortium for the Maintenance of the Text Encoding Initiative (An Agreement to Establish a Consortium for the Maintenance of the Text Encoding Initiative (March 1999)), on which basis the TEI Consortium was eventually established and incorporated as a not-for-profit legal entity at the end of the year 2000. The first members of the new TEI Board took office during January of 2001.

The TEI Consortium was established in order to maintain a permanent home for the TEI as a democratically constituted, academically and economically independent, self-sustaining, non-profit organization. In addition, the TEI Consortium was intended to foster a broad-based user community with sustained involvement in the future development and widespread use of the TEI Guidelines (Burnard (2000)).

[1] XML was originally developed as a way of publishing on the World Wide Web richly encoded documents such as those for which the TEI was designed. Several TEI participants contributed heavily to the development of XML, most notably XML's senior co-editor C. M. Sperberg-McQueen, who served as the North American editor for the TEI Guidelines from their inception until 1999.

To oversee and manage the revision process in collaboration with the TEI Editors, the TEI Board formed a Technical Council, with a membership elected from the TEI user community. The Council met for the first time in January 2002 at King's College London. Its first task was to oversee production of an XML version of the TEI Guidelines, updating P3 to enable users to work with the emerging XML toolset. This, the P4 version of the Guidelines, was published in June 2002. It was essentially an XML version of P3, making no substantive changes to the constraints expressed in the schemas apart from those necessitated by the shift to XML, and changing only corrigible errors identified in the prose of the P3 Guidelines. However, given that P3 had by this time been in steady use since 1994, it was clear that a substantial revision of its content was necessary, and work began immediately on the P5 version of the Guidelines. This was planned as a thorough overhaul, involving a public call for features and new development in a number of important areas not previously addressed including character encoding, graphics, manuscript description, biographical and geographical data, and the encoding language in which the TEI Guidelines themselves are written.

The members of the TEI Council and its associated workgroups are listed in *iii Preface and Acknowledgments*. In preparing this edition, they have been attentive to the requirements and practice of the widest possible range of TEI users, who are now to be found in many different research communities across the world, and have been largely instrumental in transforming the TEI from a grant-supported international research project into a self-sustaining community-based effort. One effect of the incorporation of the TEI has been the legal requirement to hold an annual meeting of the Consortium members; these meetings have emerged as an invaluable opportunity to sustain and reinforce that sense of community.

The present work is therefore the result of a sustained period of consultation, drafting, and revision, with input from many different experts. Whatever merits it may have are to be attributed to them; the Editors accept responsibility only for the errors remaining.

iv.3 Future Developments

The encoding recommended by this document may be used without fear that future versions of the TEI scheme will be inconsistent with it in fundamental ways. The TEI will be sensitive, in revising these Guidelines, to the possible problems which revision might pose for those who are already using this version of the Guidelines.

With TEI P5, a version numbering system is introduced: the version number has two parts, a major number and a minor, for example **1.0**. The TEI undertakes that no change will be made to the formal expression of these Guidelines (that is, a TEI schema, as defined in *23.3. Conformance*) such that documents conformant to a given major numbered release cease to be compatible with a subsequent release of the same major number. Moreover, as far as possible, new minor releases will be made only for the purpose of adding new compatible features, or of correcting errors in existing features.

The Guidelines are currently maintained as an open source (GNU General Public License) project, on the Sourceforge site `http://tei.sf.net/` from which released and development versions may be freely downloaded; notice of errors detected and enhancements requested may also be submitted at this site.

v

A Gentle Introduction to XML

The encoding scheme defined by these Guidelines is formulated as an application of the Extensible Markup Language (XML) (Bray et al. (eds.) (2006)). XML is widely used for the definition of device-independent, system-independent methods of storing and processing texts in electronic form. It is now also the interchange and communication format used by many applications on the World Wide Web. In the present chapter we informally introduce some of its basic concepts and attempt to explain to the reader encountering them for the first time how and why they are used in the TEI scheme. More detailed technical accounts of TEI practice in this respect are provided in chapters *23. Using the TEI*, *1. The TEI Infrastructure*, and *22. Documentation Elements* of these Guidelines.

Strictly speaking, XML is a metalanguage, that is, a language used to describe other languages, in this case, markup languages. Historically, the word markup has been used to describe annotation or other marks within a text intended to instruct a compositor or typist how a particular passage should be printed or laid out. Examples include wavy underlining to indicate boldface, special symbols for passages to be omitted or printed in a particular font, and so forth. As the formatting and printing of texts was automated, the term was extended to cover all sorts of special codes inserted into electronic texts to govern formatting, printing, or other processing.

Generalizing from that sense, we define markup, or (synonymously) encoding, as any means of making explicit an interpretation of a text. Of course, all printed texts are implicitly encoded (or marked up) in this sense: punctuation marks, capitalization, disposition of letters around the page, even the spaces between words all might be regarded as a kind of markup, the purpose of which is to help the human reader determine where one word ends and another begins, or how to identify gross structural features such as headings or simple syntactic units such as dependent clauses or sentences. Encoding a text for computer processing is, in principle, like transcribing a manuscript from *scriptio continua*[1]; it is a process of making explicit what is conjectural or implicit, a process of directing the user as to how the content of the text should be (or has been) interpreted.

By markup language we mean a set of markup conventions used together for encoding texts. A markup language must specify how markup is to be distinguished from text, what markup is allowed, what markup is required, and what the markup means. XML provides the means for doing the first three; documentation such as these Guidelines is required for the last.

The present chapter attempts to give an informal introduction to those parts of XML of which a proper understanding is necessary to make best use of these Guidelines. The interested reader should also consult one or more of the many excellent introductory textbooks and web sites now available on the subject.[2]

[1] In the 'continuous writing' characteristic of manuscripts from the early classical period, words are written continuously with no intervening spaces or punctuation.

[2] New textbooks about XML appear at regular intervals and to select any one of them would be invidious. A useful list of pointers to introductory web sites is available from `http://www.xml.org/xml/resources_focus_beginnerguide.shtml`; recommended online courses include `http://www.w3schools.com/xml/default.asp` and `http://www.ibm.com/developerworks/edu/x-dw-xmlintro-i.html`.

v.1 What's special about XML?

Three characteristics of XML distinguish it from other markup languages:

1. its emphasis on descriptive rather than procedural markup;
2. its notion of documents as instances of a document type;
3. its independence of any one hardware or software system.

These three aspects are discussed briefly below, and then in more depth in the remainder of this chapter.

XML is frequently compared with HTML, the language in which web pages have generally been written, which shares some of the above characteristics. Compared with HTML, however, XML has some other important features:

- XML is **extensible**: it does not consist of a fixed set of tags;
- XML documents must be **well-formed** according to a defined syntax;
- an XML document can be formally **validated** against a schema of some kind;
- XML is more interested in the meaning of data than in its presentation.

v.1.1 Descriptive markup

In a descriptive markup system, the markup codes used do little more than categorize parts of a document. Markup codes such as **<para>** or `\end{list}` simply identify a portion of a document and assert of it that 'the following item is a paragraph', or 'this is the end of the most recently begun list', etc. By contrast, a procedural markup system defines what processing is to be carried out at particular points in a document: 'call procedure PARA with parameters 42, b, and x here' or 'move the left margin 2 quads left, move the right margin 2 quads right, skip down one line, and go to the new left margin,' etc. In XML, the instructions needed to process a document for some particular purpose (for example, to format it) are sharply distinguished from the markup used to describe it.

Usually, the markup or other information needed to process a document will be maintained separately from the document itself, typically in a distinct document called a stylesheet, though it may do much more than simply define the rendition or visual appearance of a document.[3]

When descriptive markup is used, the same document can readily be processed in many different ways, using only those parts of it which are considered relevant. For example, a content analysis program might disregard entirely the footnotes embedded in an annotated text, while a formatting program might extract and collect them all together for printing at the end of each chapter. Different kinds of processing can be carried out with the same part of a file. For example, one program might extract names of persons and places from a document to create an index or database, while another, operating on the same text, but using a different stylesheet, might print names of persons and places in a distinctive typeface.

v.1.2 Types of document

A second key aspect of XML is its notion of a document type: documents are regarded as having types, just as other objects processed by computers do. The type of a document is formally defined by its constituent parts and their structure. The definition of a 'report', for example, might be that it consisted of a 'title' and possibly an 'author', followed by an 'abstract' and a sequence of one or more 'paragraphs'. Anything lacking a title, according to this formal definition, would not formally be a report, and neither would a sequence of paragraphs followed by an abstract, whatever other report-like characteristics these might have for the human reader.

[3] We do not here discuss in any detail the ways that a stylesheet can be used or defined, nor do we discuss the popular W3C Stylesheet Languages XSLT and CSS. See further Berglund (ed.) (2006), Clark (ed.) (1999), and Lie and Bos (eds.) (1999).

If documents are of known types, a special-purpose program (called a parser), once provided with an unambiguous definition of a document type, can check that any document claiming to be of that type does in fact conform to the specification. A parser can check that all elements specified for a particular document type are present and no others, that they are combined in appropriate ways, correctly ordered, and so forth. More significantly, different documents of the same type can be processed in a uniform way. Programs can be written which take advantage of the knowledge encapsulated in the document type information, and which can thus behave in a more 'intelligent' fashion.

v.1.3 Data independence

A basic design goal of XML is to ensure that documents encoded according to its provisions can move from one hardware and software environment to another without loss of information. The two features discussed so far both address this requirement at an abstract level; the third feature addresses it at the level of the strings of data characters that make up a document. All XML documents, whatever languages or writing systems they employ, use the same underlying character encoding (that is, the same method of representing as binary data those graphic forms making up a particular writing system).[4] This encoding is defined by an international standard,[5] which is implemented by a universal character set maintained by an industry group called the Unicode Consortium, and known as Unicode.[6] Unicode provides a standardised way of representing any of the many thousands of discrete symbols making up the world's writing systems, past and present.

Most modern computing systems now support Unicode directly; for those which do not, XML provides a mechanism for the indirect representation of single characters by means of their character number, known as character references; see further *v.6.1 Character References*.

v.2 Textual structures

A text is not an undifferentiated sequence of words, much less of bytes. For different purposes, it may be divided into many different units, of different types or sizes. A prose text such as this one might be divided into sections, chapters, paragraphs, and sentences. A verse text might be divided into cantos, stanzas, and lines. Once printed, sequences of prose and verse might be divided into volumes, gatherings, and pages.

Structural units of this kind are most often used to identify specific locations or refer to points within a text ('the third sentence of the second paragraph in chapter ten'; 'canto 10, line 1234'; 'page 412', etc.) but they may also be used to subdivide a text into meaningful fragments for analytic purposes ('is the average sentence length of section 2 different from that of section 5?''how many paragraphs separate each occurrence of the word *nature*? how many pages?'). Other structural units are more clearly analytic, in that they characterize a section of a text. A dramatic text might regard each speech by a different character as a unit of one kind, and stage directions or pieces of action as units of another kind. Such an analysis is less useful for locating parts of the text ('the 93rd speech by Horatio in Act 2') than for facilitating comparisons between the words used by one character and those of another, or those used by the same character at different points of the play.

In a prose text one might similarly wish to regard as units of different types passages in direct or indirect speech, passages employing different stylistic registers (narrative, polemic, commentary, argument, etc.), passages of different authorship and so forth. And for certain types of analysis (most notably textual criticism) the physical appearance of one particular printed or manuscript source may be of importance: paradoxically, one may wish to use descriptive markup to describe presentational features such as typeface, line breaks, use of whitespace and so forth.

These textual structures overlap with one other in complex and unpredictable ways. Particularly when dealing with texts as instantiated by paper technology, the reader needs to be aware of both the physical

[4] See *Extensible Markup Language (XML) 1.0*, available from `http://www.w3.org/TR/REC-xml`, Section 2.2 Characters.

[5] ISO/IEC 10646-1993 *Information Technology — Universal Multiple-Octet Coded Character Set* (UCS)

[6] See `http://www.unicode.org/`

organization of the book and the logical structure of the work it contains. Many great works (Sterne's *Tristram Shandy* for example) cannot be fully appreciated without an awareness of the interplay between narrative units (such as chapters or paragraphs) and presentational ones (such as page divisions). For many types of research, the interplay among different levels of analysis is crucial: the extent to which syntactic structure and narrative structure mesh, or fail to mesh, for example, or the extent to which phonological structures reflect morphology.

v.3 XML structures

This section describes the simple and consistent mechanism for the markup or identification of textual structure provided by XML. It also describes the methods XML provides for the expression of rules defining how units of textual structure can meaningfully be combined in a text.

v.3.1 Elements

The technical term used in XML for a textual unit, viewed as a structural component, is element. Different types of elements are given different names, but XML provides no way of expressing the meaning of a particular type of element, other than its relationship to other element types. That is, all one can say about an element called (say) <blort> is that instances of it may (or may not) occur within elements of type <farble>, and that it may (or may not) be decomposed into elements of type <blortette>. It should be stressed that XML is entirely unconcerned with the *semantics* of textual elements, because these are considered to be application dependent. It is up to the creators of XML vocabularies (such as these Guidelines) to choose intelligible element names and to define their intended use in text markup. That is the chief purpose of documents such as the TEI Guidelines. From the need to choose element names indicative of function comes the technical term for the name of an element type, which is generic identifier, or GI.

Within a marked-up text (a document instance), each element must be explicitly marked or tagged in some way. This is done by inserting a tag at the beginning of the element (a start-tag) and another at its end (an end-tag). The start- and end-tag pair are used to bracket off element occurrences within the running text, in rather the same way as different types of parentheses or quotation marks are used in conventional punctuation. For example, a quotation element in a text might be tagged as follows:

```
... Rosalind's
remarks <quote>This is the silliest stuff that ere I heard
of!</quote> clearly indicate ...
```

As this example shows, a start-tag takes the form **<quote>**, where the opening angle bracket indicates the start of the start-tag, 'quote' is the generic identifier of the element that is being delimited, and the closing angle bracket indicates the end of the start-tag. An end-tag takes an identical form, except that the opening angle bracket is followed by a solidus (slash) character, so that the corresponding end-tag is **</quote>**.[7] The material between the start-tag and the end-tag (the string of words 'This is the silliest stuff that ere I heard of' in the example above) is known as the content of the element. Sometimes there may be nothing between the start and the end-tag; in this case the two may optionally be merged together into a single composite tag with the solidus at the end, like this: **<quote/>**.

v.3.2 Content models: an example

An element may be empty, that is, it may have no content at all, or it may contain just a sequence of characters with no other elements. Often, however, elements of one type will be embedded (contained entirely) within elements of a different type.

[7] Because the opening angle bracket has this special function in an XML document, special steps must be taken to use that character for other purposes (for example, as the mathematical less-than operator); see further section *v.6.1 Character References*.

To illustrate this, we will consider a very simple structural model. Let us assume that we wish to identify within an anthology only poems, their headings, and the stanzas and lines of which they are composed. In XML terms, our document type is the anthology, and it consists of a series of poems. Each poem has embedded within it one element, a heading, and several occurrences of another, a stanza, each stanza having embedded within it a number of line elements. Fully marked up, a text conforming to this model might appear as follows:[8]

```
<anthology>
<poem>
<heading>The SICK ROSE</heading>
<stanza>
<line>O Rose thou art sick.</line>
<line>The invisible worm,</line>
<line>That flies in the night</line>
<line>In the howling storm:</line>
</stanza>
<stanza>
<line>Has found out thy bed</line>
<line>Of crimson joy:</line>
<line>And his dark secret love</line>
<line>Does thy life destroy.</line>
</stanza>
</poem>
<!-- more poems go here -->
</anthology>
```

It should be stressed that this example does *not* use the names proposed for corresponding elements elsewhere in these Guidelines: the above is thus *not* a valid TEI document.[9] It will, however, serve as an introduction to the basic notions of XML. Whitespace and line breaks have been added to the example for the sake of visual clarity only; they have no particular significance in the XML encoding itself. Also, the line

```
<!-- more poems go here   -->
```

is an XML comment and is not treated as part of the text.

As it stands, the above example is what is known as a well-formed XML document because it obeys the following simple rules:

1. there is a single element enclosing the whole document: this is known as the root element (<anthology> in our case);

2. each element is completely contained by the root element, or by an element that is so contained; elements do not partially overlap one another;

3. a tag explicitly marks the start and end of each element.

A well-formed XML document can be processed in a number of useful ways. A simple indexing program could extract only the relevant text elements in order to make a list of headings, first lines, or words used in the poem text; a simple formatting program could insert blank lines between stanzas, perhaps indenting the first line of each, or inserting a stanza number. Different parts of each poem could be typeset in different ways. A

[8] The example is taken from William Blake's *Songs of innocence and experience* (1794).

[9] The element names here have been chosen for clarity of exposition; there is, however, a TEI element corresponding to each, so that this example may be regarded as TEI conformable in the sense that this term is defined in *23.3. Conformance*.

more ambitious analytic program could relate the use of punctuation marks to stanzaic and metrical divisions.[10] Scholars wishing to see the implications of changing the stanza or line divisions chosen by the editor of this poem can do so simply by altering the position of the tags. And of course, the text as presented above can be transported from one computer to another and processed by any program (or person) capable of making sense of the tags embedded within it with no need for the sort of transformations and translations needed for files which have been saved in one or other of the proprietary formats preferred by most word-processing programs.

As we noted above, one of the attractions of XML is that it enables us to make up our own names for the elements rather than requiring us always to use names predefined by other agencies. Clearly, however, if we wish to exchange our poems with others, or to include poems others have marked up in our anthology, we will need to know a bit more about the names used for the tags. The means that XML provides for this is called a namespace. In our simple example, the tags just contain a simple name. As we shall see, it is also possible to use tags that include a qualified name, that is, a name with an optional prefix identifying the set of names to which it belongs. For example, we have defined an element <line> for the purpose of marking lines of verse. Another person might, however, define an element called <line> for the purpose of marking typographic lines, or drawn lines. Because of these different meanings, if we wish to share data it will be necessary to distinguish the two 'line' components in our marked-up texts. This is achieved by including a namespace prefix within the markup, for example like this:

```
<my:line>This is one of my lines</my:line>
<!-- ... -->
<yr:line>This is one of your lines</yr:line>
```

This feature is particularly important if we have different definitions of what a 'line' is, of course, but there are many occasions when it is useful to distinguish groups of tags belonging to different 'markup vocabularies'; we discuss this further below (*v.6.3 Namespaces*). One particularly useful namespace prefix is predefined for XML: it is `xml` and we will see examples of its use below.

Namespaces allow us to represent the fact that a name belongs to a group of names, but don't allow us to do much more by way of checking the integrity or accuracy of our tagging. Simple well-formedness alone is not enough for the full range of what might be useful in marking up a document. It might well be useful if, in the process of preparing our digital anthology, a computer system could check some basic rules about how stanzas, lines, and headings can sensibly co-occur in a document. It would be even more useful if the system could check that stanzas are always tagged **<stanza>** and not occasionally **<canto>** or **<Stanza>**. An XML document in which such rules have been checked is technically known as a valid document, and the ability to perform such validation is one of the key advantages of using XML. To carry this out, some way of formally stating the criteria for successful validation is necessary: in XML this formal statement is provided by an additional document known as a schema.[11]

v.3.3 Validating a document's structure

The design of a schema may be as lax or as restrictive as the occasion warrants. A balance must be struck between the convenience of following simple rules and the complexity of handling real texts. This is particularly the case when the rules being defined relate to texts that already exist: the designer may have only the haziest of notions as to an ancient text's original purpose or meaning and hence find it very difficult to specify consistent rules about its structure. On the other hand, where a new text is being prepared to an exact specification, for entry into a textual database of some kind for example, the more precisely stated the rules, the better they

[10] Note that this simple example has not addressed the problem of marking elements such as sentences explicitly; the implications of this are discussed in section *v.4 Complicating the issue*.

[11] The older terms Document Type Declaration and Document Type Definition, both abbreviated as DTD, may also be encountered. Throughout these Guidelines we use the term schema for any kind of formal document grammar.

can be enforced. Even in the case where an existing text is being marked up, it may be beneficial to define a restrictive set of rules relating to one particular view or hypothesis about the text — if only as a means of testing the usefulness of that view or hypothesis. A schema designed for use by a small project or team is likely to take a different position on such issues than one intended for use by a large and possibly fragmented community. It is important to remember that every schema results from an interpretation of a text. There is no single schema encompassing the absolute truth about any text, although it may be convenient to privilege some schemas above others for particular types of analysis.

XML is widely used in environments where uniformity of document structure is a major desideratum. In the production of technical documentation, for example, it is of major importance that sections and subsections should be properly nested, that cross-references should be properly resolved and so forth. In such situations, documents are seen as raw material to match against predefined sets of rules. As discussed above, however, the use of simple rules can also greatly simplify the task of tagging accurately elements of less rigidly constrained texts. By making these rules explicit, the scholar reduces his or her own burdens in marking up and verifying the electronic text, while also being forced to make explicit an interpretation of the structure and significant particularities of the text being encoded.

v.3.4 An example schema

A schema can be expressed in a number of different ways; frequently-encountered methods include the Document Type Definition (DTD) language which XML inherited from SGML; the XML Schema language (`http://www.w3.org/XML/Schema`) defined by the W3C; and the RELAX NG language (`http://relaxng.org/`) originally developed within the OASIS Technical Committee and now an ISO standard[12]. In this chapter, and throughout these Guidelines, we give examples using the 'compact syntax' of RELAX NG, but the specifications within these Guidelines are expressed in a way that is largely independent of the specific language in which a schema generated from them is expressed.[13] Although we will use the RELAX NG compact syntax for illustration in what follows, the reader should bear in mind that analogous concepts are expressed differently in other schema languages.

The following schema might be used to validate our example poem:

```
anthology_p = element anthology { poem_p+ }
poem_p = element poem { heading_p?, stanza_p+ }
stanza_p = element stanza {line_p+}
heading_p = element heading { text }
line_p = element line { text }
start = anthology_p
```

Note that this is not the only way in which a RELAX NG schema might be written;[14] we have adopted this idiom, however, because it matches that used throughout the rest of the Guidelines.

A RELAX NG schema expresses rules about the possible structure of a document in terms of patterns; that is, it defines a number of named patterns, each of which acts as a kind of template against which an input document can be matched. The meaning of a pattern is expressed in a schema by reference to other patterns, or to a small number of built-in fundamental concepts, as we shall see. In the example above, the word to the left of the equals sign is the pattern's name, and the material following it declares a meaning for the pattern. Patterns may also be of particular types; the ones that interest us here are called element patterns and attribute patterns. In this example we see definitions for five element patterns. Note that we have used similar names

[12] ISO/IEC FDIS 19757-2 Document Schema Definition Language (DSDL) -- Part 2: Regular-grammar-based validation -- RELAX NG

[13] See further *22. Documentation Elements* and *23.4. Implementation of an ODD System*. In practice, the only part of a TEI element specification not expressed using TEI-defined syntax is the content model for an element, which is expressed using the RELAX NG schema language for reasons of processing convenience. RELAX NG uses its own XML vocabulary to define content models, which is adopted by the TEI for the same purpose.

[14] For a good tutorial introduction to RELAX NG, see van der Vlist (2004).

for the pattern and the element which the pattern describes: so, for example, the line `anthology_p = element anthology {poem_p+}` defines an element pattern called `anthology_p`, the value of which defines an element called `anthology`. These naming conventions are arbitrary; we could use the same name for the pattern as for the element, since the two are syntactically quite distinct. The name, or generic identifier, of the element follows the word 'element', and the content model for the element is given within the curly braces following that. Each of these parts is discussed further below.

The last line of the schema above tells a RELAX NG validator which element (or elements) in a document can be used as the root element: in our case only <anthology>. This enables the validator to detect whether a particular document is well-formed but incomplete; it also simplifies the processing task by providing an 'entry point'.

Generic identifier

Following the word 'element' each pattern declaration gives the generic identifier (often abbreviated to GI) of the element being defined, for example *poem*, *heading*, etc. A GI may contain letters, digits, hyphens, underscore characters, or full stops, but must begin with a letter.[15] Uppercase and lowercase letters are quite distinct: an element with the GI <foo> is *not* the same as an element with the GI <Foo>; the root element of a TEI-conformant document is <TEI>, *not*<tei>.

Content model

The second part of each declaration, enclosed in curly braces, is called the content model of the element being defined, because it specifies what may legitimately be contained within it. In RELAX NG, the content model is defined in terms of other patterns, either by embedding them, or (as in our examples above) by naming or referring to them. The RELAX NG compact syntax also uses a small number of reserved words to identify other possible contents for an element, of which by far the most commonly encountered is `text`, as in this example: it means that the element being defined may contain any valid character data, but no elements. If an XML document is thought of as a structure like a family tree, with a single ancestor at the top (in our case, this would be **<anthology>**), then almost always, following the branches of the tree downwards (for example, from <anthology> to <poem> to <stanza> to <line> and <heading>) will lead eventually to `text`. In our example, <heading> and <line> are so defined, since their content models say `text` only and name no embedded elements.

Occurrence indicators

The declaration for <stanza> in the example above states that a stanza consists of one or more lines. It uses an occurrence indicator (the plus sign) to indicate how many times something matching the pattern `line_p` may be repeated. There are three occurrence indicators: the plus sign, the question mark, and the asterisk or star. The plus sign means that the pattern can match one or more times; the question mark means that it may match at most once but is not mandatory; the star means that the pattern concerned is not mandatory, but may match more than once. Thus, if the content model for <stanza> were `{line_p*}`, stanzas with no lines would be possible as well as those with more than one line. If it were `{line_p?}`, again empty stanzas would be countenanced, but no stanza could have more than a single line. The declaration for <poem> in the example above thus states that a <poem> cannot have more than one heading, but may have none, and that it must have at least one <stanza> and may have several.

Connectors

The content model `{heading_p?, stanza_p+}` contains more than one component, and thus needs additionally to specify the order in which these patterns (<heading_p> and <stanza_p>) may appear. This ordering

[15] In XML, a single colon may also appear in a GI, where it has a special significance related to the use of namespaces, as further discussed in section *v.6.3 Namespaces*. The characters defined by Unicode as combining characters and as extenders are also permitted, as are logograms such as Chinese characters.

is determined by the connector (the comma) used between its components. The comma connector indicates that the patterns concerned must appear in the sequence given. Another commonly encountered connector is the vertical bar, representing alternation. If the comma in this example were replaced by a vertical bar, then a <poem> would consist of either a heading or just stanzas – but not both!

Groups

In our example so far, the components of each content model have been either single patterns or `text`. It is quite permissible, however, to define content models in which the components are lists of patterns, combined by connectors. Such lists may also be modified by occurrence indicators and themselves combined by connectors. To demonstrate these facilities, let us expand our example to include non-stanzaic types of verse. For the sake of demonstration, we will categorize poems as one of the following: stanzaic, couplets, or blank (or stichic). A blank-verse poem consists simply of lines (we ignore the possibility of verse paragraphs for the moment),[16] so no additional elements need be defined for it. A couplet is defined as a <firstLine> followed by a <secondLine>.

```
couplet_p = element couplet {firstLine_p, secondLine_p}
```

The patterns firstLine_p and secondLine_p define elements <firstLine> and <secondLine> (which are distinguished to enable studies of rhyme scheme, for example[17]); these will have exactly the same content model as the existing <line> element. We will therefore add the following two lines to our example schema:

```
firstLine_p = element firstLine {text}
secondLine_p = element secondLine {text}
```

Next, we can change the declaration for the <poem> element to include all three possibilities:

```
poem_p = element poem
{ heading_p?, (stanza_p+ | couplet_p+ | line_p+) }
```

That is, a poem consists of an optional heading, followed by one or several stanzas, or one or several couplets, or one or several lines. Note the difference between this declaration and the following:

```
poem_p = element poem
{heading_p?, (stanza_p | couplet_p | line_p)+ }
```

The second version, by applying the occurrence indicator to the group rather than to each element within it, would allow a single poem to contain a mixture of stanzas, couplets, and lines.

A group of this kind can contain `text` as well as named elements: this combination, known as mixed content, allows for elements in which the sub-components appear with intervening stretches of character data. For example, if we wished to mark place names wherever they appear inside our verse lines, then, assuming we have also added a pattern for the <name> element, we could change the definition for <line> to

[16] It will not have escaped the astute reader that the fact that verse paragraphs need not start on a line boundary seriously complicates the issue; see further section *v.4 Complicating the issue.*

[17] This is however a rather artificial example; XPath, for example, provides ways of distinguishing elements in an XML structure by their position without the need to give them distinct names.

```
line_p = element
line { (text | name_p )* }
```

Some XML schema languages place no constraints on the way that mixed content models may be defined, but in the XML DTD language, when `text` appears with other elements in a content model: it must always appear as the first option in an alternation; it may appear once only, and in the outermost model group; and if the group containing it is repeated, the star operator must be used. Although these constraints do not apply to (for example) schemas expressed in the RELAX NG language, all TEI content models currently obey them.

Quite complex models can easily be built up in this way, to match the structural complexity of many types of text. As a further example, consider the case of stanzaic verse in which a refrain or chorus appears. Like a stanza, a refrain consists of repetitions of the line element. A refrain can appear at the start of a poem only, or as an optional addition following each stanza. This could be expressed by a pattern such as the following:

```
refrain_p = element refrain {line_p+}
poem_p = element poem   {heading_p?, ( line_p+ | (refrain_p?, (stanza_p,
refrain_p?)+ )) }
```

That is, a poem consists of an optional heading, followed by either a sequence of lines or an unnamed group, which starts with an optional refrain and is followed by one or more occurrences of another group, each member of which is composed of a stanza followed by an optional refrain. A sequence such as *refrain - stanza - stanza - refrain* follows this pattern, as does the sequence *stanza - refrain - stanza - refrain*. The sequence *refrain - refrain - stanza - stanza* does not, however, and neither does the sequence *stanza - refrain - refrain - stanza*. Among other conditions made explicit by this content model are the requirements that at least one stanza must appear in a poem, if it is not composed simply of lines, and that if there is both a heading and a stanza they must appear in that order.

Note that the apparent complexity of this model derives from the constraints expressed informally above. A simpler model, such as

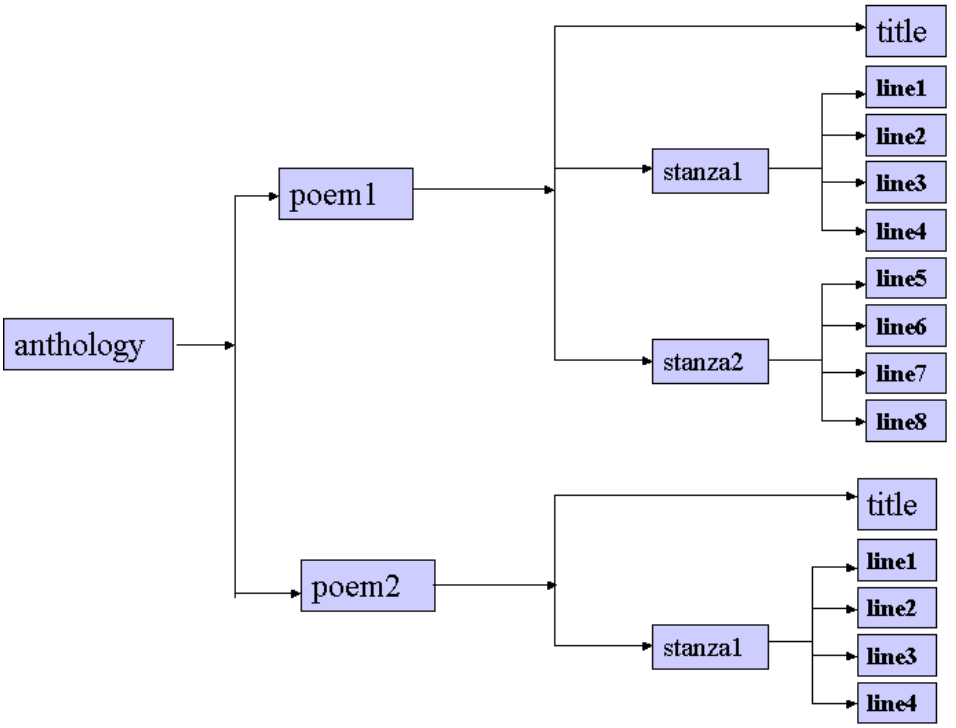
```
poem_p =
element poem {heading_p?, (line_p | refrain_p | stanza_p)+ }
```

would not enforce any of them, and would therefore permit such anomalies as a poem consisting only of refrains, or an arbitrary mixture of lines and refrains.

v.4 Complicating the issue

In the simple cases described so far, we have assumed that one can identify the immediate constituents of every element in a textual structure. A poem consists of stanzas, and an anthology consists of poems. Stanzas do not float around unattached to poems or combined into some other unrelated element; a poem cannot contain an anthology. All the elements of a given document type may be arranged into a hierarchic structure like a family tree, with a single ancestor at one end and many children (mostly the elements containing simple text) at the other. For example, we could represent an anthology containing two poems, the first of which contains two four-line stanzas and the second a single stanza, by a tree structure like the following figure:

This graphic representation of the structure of an XML document is close to the abstract model implicit in most XML processing systems. Most such systems now use a standardized way of accessing parts of an XML



document called XPath.[18] XPath gives us a non-graphical way of referring to any part of an XML document: for example, we might refer to the last line of Blake's poem as `/anthology/poem[1]/stanza[2]/line[4]`. The square brackets here indicate a numerical selection: we are talking about the fourth line in the second stanza of the first poem in the anthology. If we left out all the square-bracketted selections, the corresponding XPath expression would refer to all lines contained by stanzas contained by poems contained by anthologies. An XPath expression can refer to any collection of elements: for example, the expression `/anthology/poem` refers to all poems in an anthology and the expression `/anthology/poem/heading` refers to all their headings.

The solidus within an XPath expression behaves in much the same way as the solidus or backslash in a filename specification: it indicates that the item to the left directly contains the item to the right of it. In XPath it is also possible to indicate that any number of other items may intervene by repeating the solidus. For example, the XPath expression `/anthology/poem//line[1]` will refer to the first line of each poem in the anthology, irrespective of whether it is in a stanza.

Clearly, there are many such trees that might be drawn to describe the structure of this or other anthologies. Some of them might be representable as further subdivisions of this tree: for example, we might subdivide the lines into individual words, since in our simple example no word crosses a line boundary. Surprisingly perhaps, this grossly simplified view of what text is (memorably termed an ordered hierarchy of content objects (OHCO) view of text by Renear *et al.*[19]) turns out to be very effective for a large number of purposes. It is not, however, adequate for the full complexity of real textual structures, for which more complex mechanisms need to be employed. There are many other trees that might be drawn which do *not* fit within the anthology model which we have presented so far. We might, for example, be interested in syntactic structures or other linguistic constructs, which rarely respect the formal boundaries of verse. Or, to take a simpler example, we might want to represent the pagination of different editions of the same text.

[18] The official specification is at Clark and DeRose (eds.) (1999); many introductory tutorials are available in the XML references cited above and elsewhere on the Web: good beginners' tutorials include `http://www.w3schools.com/xpath/default.asp` and `http://www.zvon.org/xxl/XPathTutorial/`, the latter being available in several languages.

[19] See Renear et al. (1996).

In the OHCO model of text, representation of cases where different elements overlap so that several different trees may be identified in the same document is generally problematic. All the elements marked up in a document, no matter what namespace they belong to, must fit within a single hierarchy. To represent overlapping structures, therefore, a single hierarchy must be chosen, and the points at which other hierarchies intersect with it marked. For example, we might choose the verse structure as our primary hierarchy, and then mark the pagination by means of empty elements inserted at the boundary points between one page and the next. Or we could represent alternative hierarchies by means of the pointing and linking mechanisms described in chapter *16. Linking, Segmentation, and Alignment* of the Guidelines. These mechanisms all depend on the use of attributes, which may be used both to identify particular elements within a document and to point to, link, or align them into arbitrary structures.

v.5 Attributes

In the XML context, the word *attribute*, like some other words, has a specific technical sense. It is used to describe information that is in some sense descriptive of a specific element occurrence but not regarded as part of its content. For example, you might wish to add a **status** attribute to occurrences of some elements in a document to indicate their degree of reliability, or to add an **identifier** attribute so that you could refer to particular element occurrences from elsewhere within a document. Attributes are useful in precisely such circumstances.

Although different elements may have attributes with the same name (for example, in the TEI scheme, every element is defined as having an attribute named **n**), they are always regarded as different, and may have different values assigned to them. If an element has been defined as having attributes, the attribute values are supplied in the document instance as attribute-value pairs inside the start-tag for the element occurrence. An end-tag cannot contain an attribute-value specification, since it would be redundant.

The order in which attribute-value pairs are supplied inside a tag has no significance; they must, however, be separated by at least one whitespace (blank, newline, or tab) character. The value part must always be given inside matching quotation marks, either single or double[20].

For example:

```
<poem xml:id="P1" status="draft"> ... </poem>
```

Here attribute values are being specified for two attributes previously declared for the <poem> element: **xml:id** and **status**. For the instance of a <poem> in this example, represented here by an ellipsis, the **xml:id** attribute has the value **P1** and the **status** attribute has the value **draft**. An XML processor can use the values of the attributes in any way it chooses; for example, a <poem> in which the **status** attribute has the value **draft** might be formatted differently from one in which the same attribute has the value **revised**; another processor might use the same attribute to determine whether or not poem elements are to be processed at all. The **xml:id** attribute is a slightly special case in that, by convention, it is always used to supply a unique value to identify a particular element occurrence, which may be used for cross-reference purposes, as discussed further below (*v.5.2 Identifiers and indicators*).

v.5.1 Declaring attributes

Attributes are declared in a schema in the same way as elements. As well as specifying an attribute's name and the element to which it is to be attached, it is possible to specify (within limits) what kind of value is acceptable for an attribute.

[20] In the unlikely event that both kinds of quotation marks are needed within the quoted string, either or both can also be presented in escaped form, using the predefined character entities `'` or `"`

In the compact syntax of RELAX NG, an attribute is defined by means of an attribute pattern, like the following:

```
att.status = attribute status {"draft" | "revised" | "published"}
```

This defines a new pattern, called `att.status`, whose value is an attribute pattern defining an attribute named **status**. Attribute names are subject to the same restrictions as other names in XML; they need not be unique across the whole schema, however, but only within the list of attributes for a given element.

A pattern defining the possible values for this attribute is given within the curly braces, in just the same way as a content model is given for an element pattern. In this case, the attribute's value must be one of the strings presented explicitly above.

The attribute pattern definition must be included or referenced within the definition for every element to which the attribute is attached. We therefore modify the definition for the `poem_p` pattern given above as follows:

```
poem_p = element poem {att.status?, heading_p?, stanza_p+}
```

In RELAX NG, an element pattern simply includes any attribute patterns applicable to it along with its other constituents, as shown above. Attribute patterns can also be grouped and alternated in the same way as element patterns, though this particular feature is not widely used in the TEI scheme, since it is not available to the same extent in all schema languages. Because a question mark follows the reference to the `att.status` pattern in our example, a document in which the **status** attribute is not specified will still be valid; without this occurrence indicator the **status** attribute would be required.

Instead of supplying a list of explicit values, an attribute pattern can specify that the attribute must have a value of a particular type, for example a text string, a numeric value, a normalized date, etc. This is accomplished by supplying a pattern that refers to a datatype. In the example above, because a list of acceptable values is predefined, a parser can check that no <poem> is defined for which the **status** attribute does not have one of **draft**, **revised**, or **published** as its value. By contrast, with a definition such as

```
att.status =
attribute status {text}
```

a parser would accept almost any unbroken string of characters (`status="awful"`, `status="awe-ful"`, or `status="12345678"`) as valid for this attribute. Sometimes, of course, the set of possible values cannot be predefined. Where it can, as in this case, it is generally better to do so.

Schema languages vary widely in the extent to which they support validation of attribute values. Some languages predefine a small set of possibilities. Others allow the schema designer to use values from a predefined 'library' of possible datatypes, or to add their own definitions, possibly of great complexity. A 'datatype' might be something fairly general (any positive integer), something very specific or idiosyncratic (any four-character string ending with "T"), or somewhere between the two. In the RELAX NG schemas used by the TEI, general patterns have been defined for about half a dozen datatypes (using the W3C Schema Datatype Library, `http://www.w3.org/TR/xmlschema-2/`, and discussed further in *1.4.2. Datatype Macros*). In addition to the two possibilities already mentioned — plain text or an explicit list of possible strings — other datatypes likely to be encountered include the following:

boolean values must be either true or false

numeric values must represent a numeric quantity of some kind

date values must represent a possible date and time in some calendar

Two further datatypes of particular usefulness in managing XML documents are commonly known as `ID` — for identifier — and `URI` — for Universal Resource Indicator, or pointer for short. These are discussed in the next section.

v.5.2 Identifiers and indicators

It is often necessary to refer to an occurrence of one textual element from within another, an obvious example being phrases such as 'see note 6' or 'as discussed in chapter 5'. When a text is being produced the actual numbers associated with the notes or chapters may not be certain. If we are using descriptive markup, such things as page or chapter numbers, being entirely matters of presentation, will not in any case be present in the marked-up text: they will be assigned by whatever processor is operating on the text (and may indeed differ in different applications). XML therefore predefines an attribute that may be used to provide any element occurrence with a special identifier, a kind of label, which may be used to refer to it from anywhere else: since it is defined in the XML namespace, the name of this attribute is **xml:id** and it is used throughout the TEI schema. Because it is intended to act as an identifier, its values must be unique within a given document. The cross-reference itself will be supplied by an element bearing an attribute of a specific kind, which must also be declared in the schema.

Suppose, for example, we wish to include a reference within the notes on one poem that refers to another poem. We will first need to provide some way of attaching a label to each poem: this is easily done using the **xml:id** attribute. Note that not every poem need carry an **xml:id** attribute and the parser may safely ignore the lack of one in those that do not. Only poems to which we intend to refer need use this attribute; for each such poem we should now include in its start-tag some unique identifier, for example:

```
<poem xml:id="Rose"> ... </poem>
<poem xml:id="P40"> ... </poem>
<poem> ... </poem>
```

Next we need to define a new element for the cross-reference itself. This will not have any content – it is only a pointer – but it has an attribute, the value of which will be the identifier of the element pointed at. This is achieved by the following definition:

```
poemRef_p = element poemRef {attribute target {anyURI},  empty}
```

The <poemRef> element has no content, but a single attribute called **target**. The value of this attribute must be a pointer or web reference of type `anyURI`;[21] furthermore, because there is no indication of optionality on the attribute pattern, it must be supplied on each occurrence — a <poemRef> with no referent is an impossibility.

With these declarations in force, we can now encode a reference to the poem whose **xml:id** attribute specifies that its identifier is **Rose** as follows:

```
Blake's poem on the sick rose
<poemRef target='#Rose'/> ...
```

[21] The word 'anyURI' is a predefined name, used in schema languages to mean that any Uniform Resource Identifier (URI) may be supplied here. The accepted syntax for URIs is an Internet Standard, defined in `http://tools.ietf.org/html/rfc3986`. `anyURI` is one of the datatypes defined by the W3C Schema datatype library.

A processor may take any number of actions when it encounters a link encoded in this way: a formatter might construct an exact page and line reference for the location of the poem in the current document and insert it, or just quote the poem's title or first lines. A hypertext style processor might use this element as a signal to activate a link to the poem being referred to, for example by displaying it in a new window. Note, however, that the purpose of the XML markup is simply to indicate that a cross-reference exists: it does not necessarily determine what the processor is to do with it.

The target of a URI can be located anywhere: it may not necessarily be part of the same document, nor even located on the same computer system. Equally, it can be a resource of any kind, not necessarily an XML document or document fragment. It is thus a very convenient way of including references to non-XML data such as image files within a document. If, for example, we wished to include an illustration containing a reproduction of Blake's original in our anthology, the most appropriate method would probably be to define a new element called (for the sake of argument) <graphic> with a **target** attribute of datatype URI:

```
graphic_p = element graphic {att.url, empty} att.url =
attribute url {anyURI}
```

With these additions to the schema, we can now represent the location of the illustration within our text like this:

```
<poem><graphic
url="http://en.wikisource.org/wiki/Image:Blake_sick_rose.jpg"/>
</poem>
```

By providing a location from which a reproduction of the required image can be downloaded, this encoding makes it possible for appropriate software able to display the image as well as record its existence.

Attributes form part of the structure of an XML document in the same way as elements, and can therefore be accessed using XPath. For example, to refer to all the poems in our anthology whose **status** attribute has the value **draft**, we might use an XPath such as `/anthology/poem[@status='draft']`. To find the headings of all such poems, we would use the XPath `/anthology/poem[@status='draft']/heading`.

v.6 Other components of an XML document

In addition to the elements and attributes so far discussed, an XML document can contain a few other formally distinct things. An XML document may contain references to predefined strings of data that a validator must resolve before attempting to validate the document's structure; these are called entity references. They may be useful as a means of providing 'boilerplate' text or representing character data which cannot easily be keyboarded. An XML document may also contain arbitrary signals or flags for use when the document is processed in a particular way by some class of processor (a common example in document production is the need to force a formatter to start a new page at some specific point in a document); such flags are called processing instructions. And, as noted earlier, an XML document may also contain instances of elements taken from some other namespace. We discuss each of these three cases in the rest of this section.

v.6.1 Character References

As mentioned above, all XML documents use the same internal character encoding. Since not all computer systems currently support this encoding directly, a special syntax is defined that can be used to represent individual characters from the Unicode character set in a portable way by providing their numeric value, in decimal or hexadecimal notation.

For example, the character *é* is represented within an XML document as the Unicode character with hexadecimal value 00E9. If such a document is being prepared on (or exported to) a system using a different character set in which this character is not available, it may instead be represented by the character reference `é` (the `x` indicating that what follows is a hexadecimal value) or `é` (its decimal equivalent). References of this type do not need to be predefined, since the underlying character encoding for XML is always the same.

To aid legibility, however, it is also possible to use a mnemonic name (such as `eacute`) for such character references, provided that each such name is mapped to the required Unicode value by means of a construct known as an entity declaration. A reference to a named character entity always takes the form of an ampersand, followed by the name, followed by a semicolon. For example an XML document containing the string 'T&C' might be encoded as `T&C`.

There is a small set of such character entity references that do not have to be declared because they form part of the definition of XML. These include the names used for characters such as the ampersand (`amp`) and the open angle bracket or less-than sign (`lt`), which could not easily otherwise be included in an XML document without ambiguity. Other predeclared entity names are those for quotation marks (`quot` and `apos` for double and single respectively), and for completeness the closing angle bracket or greater-than sign (`gt`).

For all other named character entities, a set of entity declarations must be provided to an XML processor before the document referring to them can be validated. The declaration itself uses a non-XML syntax inherited from SGML; for example, to define an entity named eacute with the replacement value é, the declaration could have any of the following forms:

```
<!ENTITY eacute
"é">
```

or, using hexadecimal notation:

```
<!ENTITY
eacute "&#xe9;">
```

or, using decimal notation:

```
<!ENTITY eacute "&#233;">
```

Entities of this kind are useful also for string substitution purposes, where the same text needs to be repeated uniformly throughout a text. For example, if a declaration such as

```
<!ENTITY TEI "Text Encoding Initiative">
```

is included with a document, then references such as `&TEI;` may be used within it, each of which will be expanded in the same way and replaced by the string 'Text Encoding Initiative' before the text is validated.

v.6.2 Processing instructions

Although one of the aims of using XML is to remove any information specific to the processing of a document from the document itself, it is occasionally very convenient to be able to include such information — if only so that it can be clearly distinguished from the structure of the document. As suggested above, one common example is the need, when processing an XML document for printed output, to include a suggestion that the

formatting processor might use to determine where to begin a new page of output. Page-breaking decisions are usually best made by the formatting engine alone, but there will always be occasions when it may be necessary to override these. An XML processing instruction inserted into the document is one very simple and effective way of doing this without interfering with other aspects of the markup.

Here is an example XML processing instruction:

```
<?tex \newpage ?>
```

It begins with `<?` and ends with `?>`. In between are two space-separated strings: by convention, the first is the name of some processor (`tex` in the above example) and the second is some data intended for the use of that processor (in this case, the instruction to start a new page). The only constraint placed by XML on the strings is that the first one must be a valid XML name; the other can be any arbitrary sequence of characters, not including the closing character-sequence `?>`.

A construct which looks like a processing instruction (but is not) is the XML declaration which can be supplied at the beginning of an XML document, for example:

```
<?xml
version="1.0" encoding="iso-8859-1"?>
```

The XML declaration specifies the version number of the XML Recommendation applicable to the document it introduces (in this case, version 1.0), and optionally also the character encoding used to represent the Unicode characters within it. By default an XML document uses the character encoding UTF-8 or UTF-16; in this case, the 16-bit characters of Unicode have been mapped to the 8-bit character set known as ISO 8859-1; any characters present in the document but not available in the target character set will therefore need to be represented as character references (*v.6.1 Character References*). The XML declaration is purely documentary, but if it is wrong many XML-aware processors will be unable to process the associated text.

v.6.3 Namespaces

A valid XML document necessarily specifies the schema in which its constituent elements are defined. However, a well-formed XML document is not required to specify its schema (indeed, it may not even have a schema). It would still be useful to indicate that the element names used in it have some defined provenance. Furthermore, it might be desirable to include in a document elements that are defined (possibly differently) in different schemas. A cabinet-maker's schema might well define an element called <table> with very different characteristics from those of a documentalist's.

The concept of namespace was introduced into the XML language as a means of addressing these and related problems. If the markup of an XML document is thought of as an expression in some language, then a namespace may be thought of as analogous to the lexicon of that language. Just as a document can contain words taken from different languages, so a well-formed XML document can include elements taken from different namespaces. A namespace resembles a schema in that we may say that a given set of elements 'belongs to' a given namespace, or are 'defined by' a given schema. However, a schema is a set of element definitions, whereas a namespace is really only a property of a collection of elements: the only tangible form it takes in an XML document is its distinctive prefix and the identifying name associated with it.

Suppose for example that we wish to extend our anthology to include a complex diagram. We might start by considering whether or not to extend our simple schema to include XML markup for such features as arcs, polygons, and other graphical elements. XML can be used to represent any kind of structure, not simply text, and there are clear advantages to having our text and our diagrams all expressed in the same way.

Fortunately we do not need to invent a schema for the representation of graphical components such as diagrams; it already exists in the shape of the Scalable Vector Graphics (SVG) language defined by the W3C.[22] SVG is a widely used and rich XML vocabulary for representing all kinds of two-dimensional graphics; it is also well supported by existing software. Using an SVG-aware drawing package, we can easily draw our diagram and save it in XML format for inclusion within our anthology. When we do so, we need to indicate that this part of the document contains elements taken from the SVG namespace, if only to ensure that processing software does not confuse our <line> element with the SVG <line>, which means something quite different.

An XML document need not specify any namespace: it is then said to use the 'null' namespace. Alternatively, the root element of a document may supply a default namespace, understood to apply to all elements which have no namespace prefix. This is the function of the **xmlns** attribute which provides a unique name for the default namespace, in the form of a URI:

```
<anthology xmlns="http://www.example.net/anthology/ns">
</anthology>
```

In exactly the same way, on the root element for each part of our document which uses the SVG language, we might introduce the SVG namespace name:

```
<anthology xmlns="http://www.example.net/anthology/ns">
<svg xmlns="http://www.w3.org/2000/svg">
</svg>
</anthology>
```

Although a namespace name usually uses the URI (Uniform Resource Identifier) syntax, it is not treated as an online address and an XML processor regards it just as a string, providing a longer name for the namespace.

The **xmlns** attribute can also be used to associate a short prefix name with the namespace it defines. This is very useful if we want to mingle elements from different namespaces within the same document, since the prefix can be attached to any element, overriding the implicit namespace for itself (but not its children):

```
<anthology xmlns="http://www.example.net/anthology/ns"
xmlns:svg="http://www.w3.org/2000/svg">
<!-- anthology markup elements here -->
<svg:svg>
<!-- SVG markup elements here -->
</svg:svg>
<!-- more anthology markup elements here -->
</anthology>
```

There is no limit on the number of namespaces that a document can use. Provided that each is uniquely identified, an XML processor can identify those that are relevant, and validate them appropriately. To extend our example further, we might decide to add a linguistic analysis to each of the poems, using a set of elements such as <aux>, <adj>, etc., derived from some pre-existing XML vocabulary for linguistic analysis.

```
<anthology xmlns="http://www.example.net/anthology/ns"
xmlns:gram="http://www.gram.org"
xmlns:svg="http://www.w3.org/2000/svg">
```

[22] The W3C Recommendation is defined at `http://www.w3.org/Graphics/SVG/`.

```
<!-- anthology markup elements here -->
<svg:svg>
<!-- SVG markup elements here -->
</svg:svg>
<line>
<gram:itj>O</gram:itj>
<gram:nom>Rose</gram:nom>
<gram:pron>thou</gram:pron>
<gram:aux>art</gram:aux >
<gram:adj>sick</gram:adj>
</line>
</anthology>
```

Marked Sections

We mentioned above that the syntax of XML requires the encoder to take special action if characters with a syntactic meaning in XML (such as the left angle bracket or ampersand) are to be used in a document to stand for themselves, rather than to signal the start of a tag or an entity reference respectively. The predefined entities `&`, `<`, and `>` provide one method of dealing with this problem, if the number of occurrences of such things is small. Other methods may be considered when the number is large, as in an XML document like the present Guidelines, which contains hundreds of examples of XML markup. One is to label the XML examples as belonging to a different namespace from that of the document itself, which is the approach taken in the present Guidelines. Another and simpler approach is provided by one of the features inherited by XML from its parent SGML: the 'marked section'.

A marked section is a block of text within an XML document introduced by the characters `<![CDATA[` and terminated by the characters `]]>`. Between these rather strange brackets, markup recognition is turned off, and any tags or entity references encountered are therefore treated as if they were plain text. For example, when we come to write the users' manual for our anthology, we may find ourselves often producing text like the following:

```
Here is an example of the use of the <gi>line</gi> element:
<![CDATA[<line>....</line>]]>
```

v.7 Putting it all together

In this chapter we have discussed most of the components of an XML document and its associated schema. We have described informally how an XML document is represented, and also introduced one way of representing the rules a RELAX NG validator might use to validate it. In a working system, the following issues will also need to be addressed:

- how does a processor determine the schema (or schemas) that should be used to validate a given XML document instance?
- if a document contains entity references that must be processed before the document can be validated, where are those entities defined?
- an XML document instance may be stored in a number of different operating system files; how should they be assembled together?
- how does a processor determine which stylesheets it should use when processing an XML document, or how to interpret any processing instructions it contains?
- how does a processor enforce more exact validation than simple datatypes permit (for example of element content)?

Different schema languages and different XML processing systems take very different positions on all of these topics, since none of them is explicitly addressed in the XML specification itself. Consequently, the best answer is likely to be specific to a particular software environment and schema language. Since this chapter is concerned with XML considered independently of its processing environment, we only address them in summary detail here.

v.7.1 Associating entity definitions with a document instance

In *v.6.1 Character References* we introduced the syntax used for the definition of named character entities such as `eacute`, which XML inherited from SGML. Different schema languages vary in the ways they make a collection of such definitions available to an XML processor, but fortunately there is one method that all current schema languages support.

As well as, and following, the XML declaration (*v.6.2 Processing instructions*), an XML document instance may be prefixed with a special `DOCTYPE` statement. This declarative statement has been inherited by XML from SGML; in its full form it provides a large number of facilities, but we are here concerned only with the small subset of those facilities recognized by all schema languages.

Here is an example DOCTYPE statement which we might consider prefixing to the final version of our anthology:

```
<!DOCTYPE anthology [
<!ENTITY mdash "&#2014;">
<!ENTITY legalese "This document is available under a Creative Commons
Share and Enjoy Licence">
]>
```

Any XML processor encountering this statement will use it to add the two named entities it defines to those already predefined for XML. Before the document instance itself is validated, any references to these entities will be expanded to the character string given. Thus, wherever in the document instance the string `&legalese;` appears, it will be replaced by the formulation above. This makes life a little easier for those keyboarding our anthology.[23] The word `anthology` following the string DOCTYPE in this example is, of course, the name of the root element of the document to which this declaration is prefixed; however, only an XML DTD processor will take note of this fact.

v.7.2 Associating a document instance with its schema

Different schema languages adopt entirely different attitudes to this question. A document instance may be valid according to many different schemas, each appropriate to a different processing task. In RELAX NG therefore no facility for associating a particular schema with a particular instance exists: the task is regarded as a specific case of the more general issues addressed by the general architectural framework within which RELAX NG is defined: the ISO draft standard for Document Schema Definition Languages (DSDL).[24]

In W3C Schema and in the DTD schema language inherited by XML from SGML, however, a document instance can point directly to the resource or resources that may be used to validate it. In W3C Schema Language, this is usually done by means of an attribute on the root element of the document instance; for XML DTDs the `DOCTYPE` statement introduced in *v.7.1 Associating entity definitions with a document instance* is used for this purpose.

[23] And, indeed, for those responsible for deciding the licencing conditions if they change their minds later.

[24] DSDL is a project of ISO/IEC JTC 1/SC 34 WG 1, the object of which is to 'bring together different validation-related tasks and expressions to form a single extensible framework that allows technologies to work in series or in parallel to produce a single or a set of validation results. The extensibility of DSDL accommodates validation technologies not yet designed or specified.' (`http://dsdl.org`).

Fortunately, any modern XML processing software tool will provide clear ways of carrying out this task appropriate to the particular language chosen. In the interests of maximizing portability of document instances, they should contain as little processing-specific information as possible.

v.7.3 Assembling multiple resources into a single document

As we have already indicated, a single XML document may be made up of several different operating system files that need to be pulled together by a processor before the whole document can be validated. The XML DTD language defines a special kind of entity (a system entity) that can be used to embed references to whole files into a document for this purpose, in much the same way as the character or string entities discussed in *v.6.1 Character References*. Neither RELAX NG nor W3C Schema directly supports this mechanism, however, and we do not discuss it further here.

An alternative way of achieving the same effect is to use a special kind of pointer element to refer to the resources that need to be assembled, in exactly the same way as we proposed for the illustration in our anthology. The W3C Recommendation *XML Inclusions* (XInclude)[25] defines a generic mechanism for this purpose, which is supported by an increasing number of XML processors.

v.7.4 Stylesheet association and processing

As mentioned above, the processing of an XML document will usually involve the use of one or more stylesheets, often but not exclusively to provide specific details of how the document should be displayed or rendered. In general, there is no reason to associate a document instance with any specific stylesheet and the schema languages we have discussed so far do not therefore make any special provision for such association. The association is made when the stylesheet processor is invoked, and is thus entirely application-specific.

However, since one very common application for XML documents is to serve them as browsable documents over the Web, the W3C has defined a procedure and a syntax for associating a document instance with its stylesheet (see `http://www.w3.org/TR/xml-stylesheet/`). This Recommendation allows a document to supply a link to a default stylesheet and also to categorize the stylesheet according to its MIME type, for example to indicate whether the stylesheet is written in CSS or XSLT, using a specialized form of processing instruction.

Assuming therefore that we have made a CSS-conformant stylesheet for our anthology and stored it in a file called `anthology.css` which is available from the same location as the anthology itself, we could make it available over the Web simply by adding a processing instruction like the following to the anthology:

```
<?xml-stylesheet href="anthology.css"
type="text/css"?>
```

Multiple stylesheets can be defined for the same document, and options are available to specify how a web browser should select amongst them. For example, if the document also contained a directive:

```
<?xml-stylesheet href="anthology_m.css"
type="text/css" media="mobile"?>
```

a different stylesheet called `anthology_m.css` could be used when rendering the document on a handheld device such as a mobile phone.

Most modern web browsers support CSS (although the extent of their implementation varies), and some of them support XSLT.

[25] `http://www.w3.org/TR/xinclude/`.

Content validation

As we noted above, most schema languages provide some degree of datatype validation for attribute values (*v.5.1 Declaring attributes*). They vary greatly in the validation facilities they offer for the content of elements, other than the syntactic constraints already discussed. Thus, while we may very easily check that our <stanza> elements contain only <line> elements, we cannot easily check that <line> elements contain between five and 500 correctly-spelled English words, should we wish to constrain our poetry in such a way. Also, because attributes and elements are treated differently, it is difficult or impossible to express co-occurrence constraints: for example, if the **status** of a poem is **draft** we might wish to permit elements such as <editorialQuery> within its content, but not otherwise.

The XML DTD language offers very little beyond syntactic checking of element content. By contrast, a major impetus behind the design and development of the W3C schema language was the addition of a much more general and powerful constraint language to the existing structural constraints of XML DTDs. In RELAX NG the opposite approach was taken, in that all datatype validation, whether of attributes or element content, is regarded as external to the schema language. For attributes, as we have seen, RELAX NG makes use of the W3C Schema Datatype Library (but permits use of others). Because RELAX NG treats both elements and attributes as special cases of patterns, the same datatype validation facilities are available for element content as for attribute values; it is unlike other schema languages in this respect. In addition, for content validation, a different component of DSDL known as Schematron can be used. Schematron is a pattern matching (rather than a grammar-based) language, which allows us to test the components of a document against templates that express constraints such as those mentioned above.

Like other XML processors, Schematron uses XPath to identify parts of an XML document; in addition, it provides elements that describe assertions to be tested and conditions which must be validated, as well as elements to report the results of the test.

vi

Languages and Character Sets

The documents which users of these Guidelines may wish to encode encompass all kinds of material, potentially expressed in the full range of written and spoken human languages, including the extinct, the non-existent, and the conjectural. Because of this wide scope, special attention has been paid to two particular aspects of the representation of linguistic information often taken for granted: language identification, and character encoding.

Even within a single document, material in many different languages may be encountered. Human culture, and the texts which embody it, is intrinsically multilingual, and shows no sign of ceasing to be so. Traditional philologists and modern computational linguists alike work in a polyglot world, in which code-switching (in the linguistic sense) and accurate representation of differing language systems constitute the norm, not the exception. The current increased interest in studies of linguistic diversity, most notably in the recording and documentation of endangered languages, is one aspect of this long standing tradition. Because of their historical importance, the needs of endangered and even extinct languages must be taken into account when formulating Guidelines and recommendations such as these.

Beyond the sheer number and diversity of human languages, it should be remembered that in their written forms they may deploy a huge variety of scripts or writing systems. These scripts are in turn composed of smaller units, which for simplicity we term here characters. A primary goal when encoding a text should be to capture enough information for subsequent users of it correctly to identify both language, script, and constituent characters. In this chapter we address this requirement, and propose recommended mechanisms to indicate the languages, scripts and characters used in a document or a part thereof.

Identification of language is dealt with in *vi.1 Language identification*. In summary, it recommends the use of pre-defined identifiers for a language where these are available, as they increasingly are, in part as a result of the twin pressures of an increasing demand for language-specific software and an increased interest in language documentation. Where such identifiers are not available or not standardized, these Guidelines recommend a way of documenting language identifiers and their significance, in the same way as other metadata is documented in the TEI Header.

Standardization of the means available to represent characters and scripts has moved on considerably since the publication of the first version of these Guidelines. At that time, it was essential to explicitly document the characters and encoded character sets used by almost any digital resource if it was to have any chance of being usable across different computer platforms or environments, but this is no longer the case. With the availability of the Unicode standard, almost 100,000 different characters representing almost all of the world's current writing systems are available and usable in any XML processing environment without formality. Nevertheless, however large the number of standardized characters, there will always be a need to encode documents which use non-standard characters and glyphs, particularly but not exclusively in historical material. Furthermore, the full potential of Unicode is still not yet realised in all software which users of the Guidelines are likely

to encounter. The second part of this chapter therefore discusses in some detail the concepts and practice underlying this standard, and also introduces the methods available for extending beyond it, which are more fully discussed in *5. Representation of Non-standard Characters and Glyphs*.

vi.1 Language identification

Identification of the language a document or part thereof is written in is a crucial requirement for many envisioned usages of an electronic document. The TEI therefore accomodates this need in the following way:

- A global attribute **xml:lang** is defined for all TEI elements. Its value identifies the language and writing system used.
- The TEI Header has a section set aside for the information about the languages used in a document: see further *2.4.2. Language Usage*.

The value of the attribute **xml:lang** identifies the language using a coded value. For maximal compatibility with existing processes, modelling this value in the following way is recommended (this parallels the modelling of **xml:lang**):

- The identifier for the language should be constructed as in *Best Current Practice 47*[1]. This *same* identifier has to be used to identify the corresponding <language> element in the TEI header, if one is present.

The first part of BCP 47 is called *Tags for Identifying Languages*[2], and proposes the following mechanism for constructing an identifier (tag) for languages as administered by the Internet Assigned Numbers Authority (IANA). The tag is assembled from a sequence of subtags separated by the hyphen (-, U+002D) character. It gives the language (possibly further identified with a sublanguage), a script and a region for this language, each possibly followed by a variant subtag.

- The identifier consists of at least one 'primary' subtag, it may be followed by one or more 'extended' subtags.
- Languages are identified by a language subtag, which may be a two letter code taken from ISO 639-1 or a three letter code taken from ISO 639-2.
- ISO 639-2 reserves for private use codes in the range 'qaa' to 'qtz'. These codes should be used for non-registered language subtags.
- A single letter primary subtag "x" indicates that the whole language tag is privately used.
- Extended language subtags must begin with the letter "s". They must follow the primary subtag and precede subtags that do define other properties of the language. The order is significant.
- 4 character subtags are interpreted as script identifiers taken from ISO 15924
- Region subtags can be either two letter country codes taken from ISO 3166 (with exceptions) or 3 digit codes from the UN Standard Country Codes for Statistical Use.
- Variant subtags may follow any of the above, but must precede private use extensions.
- Private use extensions are separated from the other subtags by the single letter subtag "x", which must be followed by at least one subtag. They might consist of several subtags separated with "-", but may not exceed a length of 32 characters.

Examples of language tags

- **Simple language subtag**
  - de (German)
  - ja (Japanese)

[1] Currently BCP 47 comprises two Internet Engineering Task Force documents, referred to separately as RFC 4646 and RFC 4647; over time, other IETF documents may succeed these as the best current practice.

[2] Phillips, Addison and Davis, Mark, *Tags for Identifying Languages*2006-09: `http://tools.ietf.org/html/bcp47`

  - zh (Chinese)

- **Language subtag plus Script subtag**
  - zh-Hant (Traditional Chinese)
  - en-Latn (English written in Latin script)
  - sr-Cyrl (Serbian written with Cyrillic script)

- **Language-Script-Region**
  - zh-Hans-CN (Simplified Chinese for the PRC)
  - sr-Latn-891 (Serbian, Latin script, Serbia and Montenegro)

- **Language-Region**
  - zh-SG (Chinese for Singapore)
  - de-DE (German for Germany)

- **Other**
  - zh-CN (Chinese in China, no script given)
  - zh-Latn (Chinese transcribed in the Latin script)

- **Extended:**
  - de-CH-x-phonebook (phonebook collation for Swiss German)
  - zh-s-nan (the Southern Min language of the macrolanguage Chinese)
  - zh-s-nan-Hans-CN (the Southern Min language of the macrolanguage Chinese as spoken in China written in simplified Characters)
  - zh-Latn-x-pinyin (Chinese transcribed in the Latin script using the Pinyin system)

It should be noted that capitalization given here follows established convention (e.g. capital letters for country codes, small letters for language codes), but BPC 47 does not ascribe any meaning to differences in capitalization.

As can be seen, both BPC 47 and ISO 639-2 provide extensions that can be employed by private convention. The constructs mentioned above can thus be used to generate identifiers for any language, past and present, in any used in any area of the world. If such private extensions are used within the context of the TEI, they should be documented within the <language> element of the TEI header, which might also provide a prose description of the language described by the language tag.

While language, region and script can be adequately identified using this mechanism, there is only very rough provision to express a dimension of time for the language of a document; those codes provided (e.g. `grc` for 'Greek, Ancient (to 1453)' in ISO 639-2) might not reflect the segments appropriate for a text at hand. Text encoders might express the time window of the language used in the document by means of the extension mechanism defined in BCP 47 and relate that to a <date> element in the corresponding <language> section of the TEI header.

Equivalences to language identifiers by other authorities can be given in the <language> section as well, but no formal mechanism for doing so has been defined.

The scope of the language identification is extending to the whole subtree of the document anchored at the element that carries the **xml:lang** attribute, including all elements and all attributes where a language might apply.[3]

[3] This will exclude all attributes where a non-textual datatype has been specified, for example tokens, boolean values or predefined value lists.

vi.2 Characters and Character Sets

All document encoding has to do with representing one thing by another in an agreed and systematic way. Applied to the smallest distinctive units in any given writing system, which for the moment we may loosely call 'characters', such representation raises surprisingly complex and troublesome issues. The reasons are partly historical and partly to do with conceptual unclarities about what is involved in identifying, encoding, processing and rendering the characters of a natural language.

vi.2.1 Historical considerations

When the first methods of representing text for storage or transmission by machines were devised, long before the development of computers, the overriding aim was to identify the smallest set of symbols needed to convey the essential semantic content, and to encode that symbol set in the most economical way that the storage or transmission media allowed. The initial outcome were systems that encoded only such content as could be expressed in uppercase letters in the Latin script, plus a few punctuation marks and some 'control characters' needed to regulate the storage and transmission devices. Such encodings, originally developed for telegraphy, strongly influenced the way the pioneers of computing conceived of and implemented the handling of text, with consequences that are with us still.

For many years after the invention of computers, the way they represented text continued to be constrained by the imperative to use expensive resources with maximal efficiency. Even when storage and processing costs began their dramatic fall, the Anglo-centric outlook of most hardware designers and software engineers hampered initiatives to devise a more generous and flexible model for text representation. The wish to retain compatability with 'legacy' data was an additional disincentive. Eventually, tension in East Asia between commitment to technological progress and the inability of existing computers to cope with local writing systems led to decisive developments. Japanese, Korean and Chinese standards bodies, who long before the advent of computers had been engaged in the specification of character sets, joined with computer manufacturers and software houses to devise ways of mapping those character sets to numeric encodings and processing the resulting text data.

Unfortunately, in the early years there was little or no co-ordination among either the national standards bodies or the manufacturers concerned, so that although commercial necessity dictated that these various local standards were all compatible with the representation of US-American English, they were not straightforwardly compatible with one another. Even within Japan itself there emerged a number of mutually incompatible systems, thanks to a mixture of commercial rivalry, disagreements about how best to manage certain intractable problems, and the fact that such pioneering work inevitably involved some false starts, leading to incompatibilities even between successive products of the same bodies. Roughly at the same time, and for similar reasons, multiple and incompatible ways of representing languages that use Cyrillic scripts were devised, along with methods of encoding ancient writing systems which inevitably could not aim for compatibility with other writing systems apart from basic Latin script. Many of the earliest projects that fed into the TEI were shaped in this developmental phase of the computerised representation of texts, and it was also the context in which SGML was devised and finalized.

SGML had of necessity to offer ways of coping with multiple writing systems in multiple representations; or rather, it provided a framework within which SGML-compliant applications capable of handling such multiple representations might be developed by those with sufficient financial and personnel resources (such as are seldom found in academia). Earlier editions of these Guidelines offered advice on character set and writing system issues addressed to the condition of those for whom SGML was the only feasible option. That advice must now be substantially altered because of two closely-related developments: the availability of the ISO/Unicode character set as an international standard, and the emergence of XML and related technologies which are committed to the theory and practice of character representation which Unicode embodies.

vi.2.2 Terminology and key concepts

Before the significance of Unicode and the implications of the association between XML and Unicode can be adequately explained, it is necessary to clarify some key concepts and attempt to establish an adequately precise terminology for them.

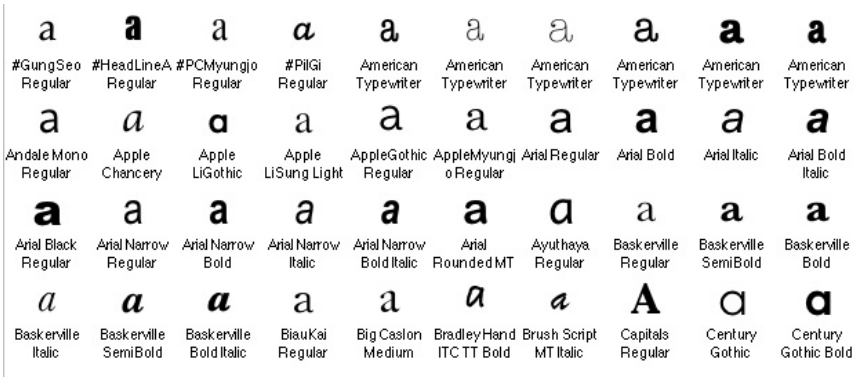


Figure vi.1: Examples of the small latin *a* rendered with different fonts.

The word 'character' will not of itself take us very far towards greater terminological precision. It tends to be used to refer indiscriminately both to the visible symbol on a page and to the letter or ideograph which that symbol represents, two things that it is essential to keep conceptually distinct. The visible symbol obviously has some aspects by which we interpret it as representing one character rather than another; but its appearance may also be significantly determined by features that have no effect on our notion of which character in a writing system it represents. A familiar instance is the lowercase *a*, which in printed texts may be represented either by a 'single storey' symbol (cf. figure 1 in the examples from Baskerville SemiBold or Century) or by a 'two storey' version (as in figure 1 in the examples from ArialRegular or Andale Mono Regular). We say that the single and double-storey symbols both represent one and the same the same *abstract character a* using two different *glyphs*. Similarly, an uppercase *A* in a serif typeface has additional strokes that are absent from the same letter when printed using a sans-serif typeface, so that once again we have differing glyphs standing for the same abstract character. In figure 1 there is even a font, Captials Regular, in which the glyph for the lowercase letter *a* looks like a typical glyph for the character uppercase *A*. The distinction between abstract characters and glyphs is fundamental to all machine processing of documents.

In most scholarly encoding projects, the accurate recording of the abstract characters which make up the text is of prime importance, because it is the essential prerequisite of digitizing and processing the document without semantic loss. In many cases (though there are important exceptions, to be touched on shortly) it may not be necessary to encode the specific glyphs used to render those abstract characters in the original document. An encoding that faithfully registers the abstract characters of a document allows us to search and analyse our document's content, language and structure and access its full semantics. That same encoding, however, may not contain sufficient information to allow an exact visual representation of the glyphs in the source text or manuscript to be recreated.

The importance of this distinction between information content and its visual representation is not always immediately apparent to people unused to the specific complexities of text handling by machine. Such users tend to ask first what (in order of conceptual priority) should actually be their very last question: how do I get a physical image that looks like character x in my source document to appear on to the screen or the output page? Their first question should in fact be: how can I get an abstract representation of character x into my encoded document in a way that will be universally and unambiguously identifiable, no matter what it happens to look like in printout or on any particular display? And occasionally the response they receive as a result of their misguided initial question is a custom 'solution' that satisfies their immediate rendering wishes at the price of

making their underlying document unintelligible to other users (or even to the original user in other times and places) because it encodes the abstract character in an idiosyncratic way.

That said, there will certainly be documents or projects where it is a matter of scholarly significance that the compositor or scribe chose to represent a given abstract character using one particular glyph or set of strokes rather than a semantically-equivalent but visually distinct alternative, and in that case the specific appearance of the form will have to be encoded on one way or another. But that encoding need not (and in most cases will not) involve a notation that visually resembles the original, any more than italicised text in an original document will be represented by the use of italic characters in the encoded version.

A collection of the abstract characters needed to represent documents in a given writing system is known as a character set, and the character set or character repertoire of a processing or rendering device is the set of abstract characters that it is equipped to recognise and handle appropriately. There is, however, a subtle distinction between these two parallel uses of the same term, involving one more key concept which it is essential to grasp. The character set of a document (or the writing system in which it is recorded) is purely a collection of abstract characters. But the character set of a computing device is a set of abstract characters which have been mapped in a well-defined way to a set of numbers or code points by which the device represents those abstract characters internally. It can therefore be referred to as a coded character set, meaning a set of abstract characters each of which has been assigned a numerical code point (or in some instances a sequence of code points) which unambiguously identifies the character concerned.

It is now possible to use this terminology to say what Unicode is: it is a coded character set, devised and actively maintained by an international public body, where each abstract character is identified by a unique name and assigned a distinctive code point.[4] Unicode is distinguished from other, earlier and co-existing coded character sets by its (current and potential) size and scope; its built-in provision for (in practical terms) limitless expansion; the range and quality of linguistic and computational expertise on which it draws; the commitment in principle (and to an increasing degree in practice) to implement it by all important providers of hardware and software worldwide; and the stability, authority and accessibility it derives from its status as an international public standard.

vi.2.3 Abstract characters, glyphs and encoding scheme design

The distinction between abstract characters and glyphs can be crucial when devising an encoding scheme. Users performing text retrieval, searching or concordancing will expect the system to recognise and treat different glyphs as instances of the same character; but when perusing the text itself they may well expect to see glyph variants preserved and rendered. When encoding a pre-existing text, the encoder must determine whether a particular letter or symbol is a character or a glyphic variant. A detailed model of the relationship between characters and glyphs has been developed within the Unicode Consortium and an ISO work group (ISO/IEC JTC1 SC2/WG2). Its report (Unicode Technical Report 17: Character Encoding Model) will form the base for much future standards work.

The model makes explicit the distinction between two different properties of the components of written language:

- their content, i.e. its meaning and phonetic value (represented by a character)
- their graphical appearance (represented by a glyph)

When searching for information, a system generally operates on the content aspects of characters, with little or no attention to their appearance. A layout or formatting process, on the other hand, must of necessity be concerned with the exact appearance of characters. Of course, some operations (hyphenation for example) require attention to both kinds of feature, but in general the kind of text encoding described in these Guidelines tends to focus on content rather than appearance (see further 6.3 Highlighting and Quotation).

[4] Although only Unicode is mentioned here explicitly, it should be noted that the character repertoire and assigned code points of Unicode and the ISO standard 10646 are identical and maintained in a way that ensures this continues to be the case.

An encoder wishing to record information about which glyphs are present in a given document may do so at either or both of two levels:

- the level of character encoding, using an appropriate Unicode code point to represent the glyph concerned
- the markup level, with the glyph indicated via appropriate elements and/or attributes

The encoding practice adopted may be guided by, among other things, an assessment of the most frequent uses to which the encoded text will be put. For example, if recognition of identical characters represented by a variety of glyphs is the main priority, it may be advisable to represent the glyph variations at markup level, so that the character value can be immediately exposed to the indexing and retrieval software. Plainly, an encoding project will need to consider such issues carefully and embody the outcome of their deliberations in local manuals of procedure to ensure encoding consistency. Using Unicode code points to represent glyph information requires that such choices be documented in the TEI Header. Such documentation does cannot of itself guarantee proper display of the desired glyph but at least makes the intention of the encoder discoverable.

At present the Unicode Standard does not offer detailed specifications for the encoding of glyph variations. These Guidelines do give some recommendations; some discussion of related matters is given in Chapter 18 Transcription of Primary Sources, and Chapter 25 Representation of non-standard Characters and Glyphs offers some features for the definition of variant glyphs.

vi.2.4 Entry of characters.

Text characters may be entered into a document using any of three methods, in any convenient combination. First, where suitable input facilities make this possible, the characters concerned may be entered directly into the document, either by normal keystrokes or by the use of one of the Input Method Editors (IMEs) commonly used for the entry of ideographic characters. This is most likely to be convenient where the display used for text entry and/or the printer used to produce output for proofreading purposes is capable of rendering the characters concerned using correct and readily identifiable glyphs. Where such easily checkable rendering is not available, or where there is no suitable method of inputting certain characters directly, they may be input by one of two possible forms of indirect notation or 'reference'.

The first form of reference is a Numeric Character Reference (NCR), which takes the general form `&#D;` where `D` is an integer representing the code point of the character in base 10, or `&#xH;`, where `H` is the code point in hexadecimal notation. This has the advantage that no declaration of what this notation means is required anywhere in the document instance or its associated schema. Every XML processor is capable of recognising NCRs and replacing them with the required code point value without needing access to any additional data. The disadvantage of NCRs as a means of entering, representing and proofing character data is that most human beings find them anything but 'readable' and it is all too easy for the wrong character to be entered in error and retained undetected.

The second form of reference is a Character Entity Reference (though, as explained below, this should not be taken to imply that such entities constitute a 'type' that could be distinctively recognised by a processing system). Character entity references can (and indeed should) have names whose significance is apparent to humans, but each and every entity name has to be associated with its replacement (which as explained below should be a character value, possibly in the form of a NCR) via a formal declaration in the document's internal or external subset. For a large number of characters defined by Unicode and commonly used in documents, there are ISO entity sets declaring mnemonic names which should be used wherever feasible: XML compatible character entity declarations using ISO names and suitable for inclusion into the subset are available on the TEI web sites.

Where characters are not defined in Unicode and so have to be assigned both a local code point and a local entity name of the project's choosing (see *Non Unicode characters in XML documents* below) it is highly desirable to follow the same nomenclature principles as ISO and to emulate the practice in the ISO character entity declarations of appending a string giving the character a unique descriptive name as a comment to the

actual entity declaration. In addition, where different groups or projects are working on texts with geographical, historical, linguistic or other similarities that give rise to common issues of character encoding, it is highly advisable in the interests of consistency that they should consult one another when devising entity names. The TEI mailing list may provide a suitable first point of contact for such consultations. Further advice on the matter of locally-defined characters is contained in Chapter 25 Representation of non-standard Characters and Glyphs.

vi.2.5 Output of characters

Rendering of the encoded text is a complicated process that depends largely on the purpose, external requirements, local equipment and so forth, it is thus outside the scope of coverage for these Guidelines.

It might however nevertheless be helpful to put some of the terminology used for the rendering process in the context of the discussion of this chapter. As was mentioned above, Unicode encodes abstract characters, not specific glyphs. For any process that makes characters visible, however, concrete, specifically designed glyph shapes have to be used. For a printing process, for example, these shapes describe exactly at which point ink has to be put on the paper and which areas have to be left blank. If we want to print a character from the Latin script, besides the selection of the overall glyph shape, this process also requires that a specific weight of the font has been selected, a specific size and to what degree the shape should be slanted. Beyond individual characters, the overall typesetting process also follows specific rules of how to calculate the distance between characters, how much whitespace occurs between words, at which points line breaks might occur and so forth.

If we concern ourselves only with the rendering process of the characters themselves, leaving out all these other parameters, we will realize that of all the information required for this process, only a small amount will be drawn from the encoded text itself. This information is the code point used to encode the character in the document. With this information, the font selected for printing will be queried to provide a glyph shape for this character. Some modern font formats (e.g. OpenType) do implement a sophisticated mapping from a code point to the glyph selected, which might take into account surrounding characters (to create ligatures where necessary) and the language or even area this character is printed for to accomodate different typesetting traditions and differences in the usage of glyphs.

A TEI document might provide some of the information that is required for this process for example by identifying the linguistic context with the **xml:lang** attribute. The selection of fonts and sizes is usually done in a stylesheet, while the actual layout of a page is determined by the typesetting system used. Similarily, if a document is rendered for publication on the Web, information of this kind can be shipped with the document in a stylesheet[5].

vi.2.6 Unicode and XML

The devisers of the XML standard took the view that Unicode should be the only means of representing abstract characters which conformant XML processors were obliged to support. That certainly does not preclude the use of other character encoding schemes or character sets in documents which are to be handled by XML processors, but it does mean that all the abstract characters which are encoded as characters (as distinct from being represented indirectly via markup) in an XML document must either possess an assigned code point within the public Unicode standard, or be assigned a code point devised by and specific to the local project, taken from a reserved range set aside by the standard expressly for this purpose, the so-called Private Use Areas or PUAs. For the vast majority of projects to which these Guidelines are applicable, the Unicode standard will already offer code points for all the abstract characters their documents employ, and so the requirement that all such characters should be resolvable by XML processors to Unicode code points will not involve any definition or use of PUA code points. Indeed, such projects are not obliged by their choice of XML to use Unicode in their

[5]The World Wide Web Consortium provides recommendations for two standard stylesheet languages: either CSS or XSL could be used for this purpose.

documents. Provided they correctly declare at the requisite points any non-Unicode coded character set they may use, ensure that all their XML processors support their declared encoding, and then consistently employ that encoding in strict conformity with their declarations, they need not consciously concern themselves with Unicode unless and until they feel it is appropriate to do so.

Non-Unicode character sets and XML processors

There are, however, strict limits to the way conformant XML processors handle documents whose character set is not Unicode, and unless these limits are understood it is likely that projects not yet ready to commit to Unicode across the board will run into unexpected and baffling problems as they attempt to operate with their legacy character encodings. First, it must be repeated that nothing in the XML standard *requires* conformant processors to handle non-Unicode documents. But even if there were any actual processors which on that basis refused to process non-Unicode documents, that would not limit their usefulness as severely as might at first appear. The reason is that there is a way of internally representing Unicode code points (explained further *Encoding errors related to UTF-8* below) where there is no detectable difference between a document which is actually encoded in ASCII employing only 7-bit values and one which is encoded in Unicode but which happens to contain only the abstract characters encompassed by the 7-bit ASCII standard. And the XML standard specifies that this way of representing Unicode is the one which processors must assume as the default for any document that does not explicitly declare an encoding. At a stroke, this provision ensures that all pure 7-bit ASCII encoded documents can be processed without further ado by all conformant XML processors. Add to this the provision, also within the XML standard, that allows any Unicode code point to be indirectly specified using only 7-bit ASCII characters via a Numeric Character Reference (NCR), and the upshot is that all documents in non-Unicode encodings which can be pre-processed to rewrite any characters outside the 7-bit ASCII range as Unicode code points in NCR notation (a simple batch procedure for which software is readily available) can be handled even by processors which have no inbuilt support for any encoding other than Unicode.

In fact, every XML processor so far released has implemented methods, specified in the standard though not mandatory, which allow the processing of documents in at least some non-Unicode character sets. Such processors include in their documentation a statement of the non-Unicode encodings they support, and the use of such an encoding must be declared to the processor in the correct way.

To avoid confusion when taking advantage of such encoding support, it is first of all essential to grasp that an encoding declaration in an XML document is indeed simply a declaration: it is not an incantation that magically converts the document that follows into the encoding concerned. It is a common error to think that simply declaring a document's encoding to be, say ISO-8859-1 (or for that matter UTF-8 or UTF-16, the representations of Unicode for which support is mandatory) is sufficient to 'make it so'. Such a declaration is useless unless the document that follows actually is encoded strictly in conformance with the declaration. Some of the circumstances in which that may not in fact be the case are outlined in *vi.2.9 Issues arising from the internal representations of Unicode* below. Secondly, an encoding declaration does not somehow switch an XML processor into a mode where it works entirely in the declared encoding for as long as the declaration is in scope. On the contrary, all it does is instruct the processor to pass its input through a filter that immediately converts all the code points in the declared encoding into their Unicode counterparts; from that point onwards the document as seen by all subsequent stages of processing is actually in Unicode, even though that may not be apparent to the user. Thirdly, this invariable internal conversion has a crucial consequence: the fact that a processor can successfully accept a document in a non-Unicode encoding does not mean that it will necessarily convert any output it may produce back into the declared input encoding. Internally, the document has been converted to and processed in Unicode, and there is nothing in the XML standard that requires the reverse conversion to be performed at the output stage. Most processors go beyond the standard by offering a facility to output in various encodings: but whether it is available and how to use it must be ascertained from the processor's documentation. Should it be unavailable or unreliable, the output may need to be post-processed

through a character convertor to restore the original encoding, and again such software is freely available and easy to use.

Non Unicode characters in XML documents

In the cases considered in the preceding section, there was a suitable Unicode code point corresponding to each abstract character contained in the non-Unicode character set of the input document. In such instances, the mandatory internal conversion to Unicode carried out by the processor can be more or less transparent to a user who wishes to continue to work with a non-Unicode character set. Things become rather different when the non-Unicode character set contains abstract characters for which there is no code point in the Unicode standard, or when a project that is attempting to work in Unicode throughout finds that it needs to represent abstract characters not currently provided for in the Unicode standard. Here, a significant difference between SGML and XML emerges in a rather troublesome way.

Following their agenda to devise a subset of SGML that would be significantly easier to implement, the authors of the XML specification decided that one particular type of entity available in SGML, known as an internal SDATA entity, should not be carried over into XML. It would be idle to question that decision here, but its consequences for the handling of abstract characters for which there is no Unicode definition were significant.

The procedures recommended in earlier versions of these Guidelines for encoding, processing and exchanging what we might call locally defined abstract characters were reliant on the availability of entities declared as of type SDATA, but that type is not supported in XML, and there is therefore no ready equivalent for XML-based projects to the recommendations previously offered.[6] Entities in XML are really only of two basic types, parsed and unparsed. Unparsed entities are of no relevance here. References to parsed entities in an XML document result in only one kind of behaviour: when they appear in the parser's input stream, the parser expects to be able to resolve them by locating a declaration in the document's internal or external subset which maps the entity name to its replacement text. The parser then inserts that replacement text into the document in place of the entity reference, which is discarded without trace. The act of replacement is not notified to the application, except where it fails because the entity is undeclared or the declaration is in some way defective (in which case the parser signals a fatal error and stops.)

Though for explanatory convenience much XML-related documentation, including these Guidelines, refers specifically to Character Entities and Character Entity References, a character entity in XML is not a distinct 'type' in the sense that 'type' is understood in Computer Science terminology, for example when referring to the type of an attribute. Hence there is no way in which editing or other software can check that the replacement to be inserted is indeed a single character or its equivalent rather than an arbitrary chunk of text, possibly including markup. A character entity is simply a general entity whose replacement text happens to be declared as a character value or a NCR representing that value. This has two important consequences if it is proposed to use such an entity reference to stand for a character that has no Unicode equivalent. First, the entity name reference will disappear at an early stage in the parse and be replaced by the declared value of the entity, so that no processing which requires access in the parsed document to the entity reference as originally entered is possible. Secondly, if a character entity is to be used as a true equivalent to a normal character, and consequently be employed at all points in a document where a single character could legitimately occur (apart from in element and attribute names, where no references of any kind are allowed) then it is essential that its replacement value indeed be pure character data. If the replacement value of the entity were to contain any markup, or a processing instruction, there would be many places in a document where simple character data would be legitimate, but where the substitution of markup or some other replacement could cause the

[6] In essence, when an SGML parser encounters a reference to an entity of type SDATA, it supplies to the application which it is servicing the name of that entity, as found in the document, plus a pointer to a location somewhere on the local system, and what is present at that location may in turn allow or instruct the application to do one of a number of things, including looking up the entity name in a table and deriving information about the referenced entity which can trigger specific behaviours in the application appropriate to the processing of that abstract character. There is however no way to make an XML parser do anything of the kind in response to an entity reference.

document to become invalid or malformed. Taken together, these considerations mean that the transparent use of a CER to stand for a non-Unicode character in an XML document is simply not possible.

vi.2.7 Special aspects of Unicode character definitions

Compatibility characters

The principles of Unicode are judiciously tempered with pragmatism. This means, among other things, that the actual repertoire of characters which the standard encodes, especially those parts dating from its earlier days, include a number of items which on a strict interpretation of the Unicode Consortium's theoretical approach should not have been regarded as abstract characters in their own right. Some of these characters are grouped together into a code-point regions assigned to compatibility characters. Ligatures are a case in point. Ligatures (.e.g. the joining of adjacent lowercase letters ‘s’ and ‘t’ or ‘f’ and ‘i’ in Latin scripts, whether produced by a scribal practice of not lifting the pen between strokes or dictated by the aesthetics of a type design) are representational features with no added semantic value beyond that of the two letters they unite (though for historians of typography their presence and form in a given edition may be of scholarly significance). However, by the time the Unicode standard was first being debated, it had become common practice to include single glyphs representing the more common ligatures in the repertoires of some typesetting devices and high-end printers, and for the coded character sets built into those devices to use a single code point for such glyphs, even though they represent two distinct abstract characters. So as to increase the acceptance of Unicode among the makers and users of such devices, it was agreed that some such pseudo-characters should be incorporated into the standard. Nevertheless, if a project requires the presence of such ligatured forms to be encoded, this should normally be done via markup, not by the use of a compatibility character. That way, the presence of the ligature can still be identified (and if desired, rendered visually) where appropriate, but indexing and retrieval software will treat the code points in the document as a simple sequential occurrence of the two constituent characters concerned and so correctly align their semantics with non-ligatured equivalents. Such ligatures should not be confused with digraphs (usually) indicating diphthongs, as in the French word "cœur". Digraphs are atomic orthographic units representing abstract characters in their own right, not purely glyphic amalgamations, and indexing and retrieval software must treat them as such. Where a digraph occurs in a source text, it should normally be encoded using the appropriate code point for the single abstract character which it indeed represents, either by direct entry of the character concerned of through the appropriate CER or NCR.

Precomposed and combining characters and normalization

The treatment of characters with diacritical marks within Unicode shows a similar combination of rigour and pragmatism. It is obvious enough that it would be feasible to represent many characters with diacritical marks in Latin and some other scripts by a sequence of code points, where one code point designated the base character and the remainder represented one or more diacritical marks that were to be combined with the base character to produce an appropriate glyphic rendering of the abstract character concerned. From its earliest phase, the Unicode Consortium espoused this view in theory but was prepared in practice to compromise by assigning single code points to precomposed characters which were already commonly assigned a single distinctive code point in existing encoding schemes. This means, however, that for quite a large number of commonly-occurring abstract characters, Unicode has two different, but logically and semantically equivalent encodings: a precomposed single code point, and a code point sequence of a base character plus one or more combining diacritics. Scripts more recently added to Unicode no longer exhibit this code-point duplication (in current practice no new precomposed characters are defined where the use of combining characters is possible) but this does nothing to remove the problem caused by the duplications permanently embodied in older strata of the character set. Together with essentially analogous issues arising from the encoding of certain East Asian ideographs, this duplication gives rise to the need to practice normalization of Unicode documents. Normalization is the process of ensuring that a given abstract character is represented in one way only in a given

Unicode document or document collection. The Unicode Consortium provides four standard normalization forms, of which the Normalization Form C (NFC) seems to be most appropriate for text encoding projects. The World Wide Web Consortium has produced a document entitled *Character Model for the World Wide Web 1.0*[7], which among other things discusses normalization issues and outlines some relevant principles. An authoritative reference is Unicode Standard Annex #15 *Unicode Normalization Forms*[8]. Individual projects will have to decide how far their decisions on normalization need be influenced by the fact that at present, by no means all hardware or software can correctly render (or even consistently identify) abstract characters encoded using combining symbols. It should be noted however, that normalization as discussed in the documents above does not cover the problems mentioned above with East-Asian characters, except for issues connected with composed characters in Hangul.

It is important that every Unicode-based project should agree on, consistently implement and fully document a comprehensive and coherent normalization practice. As well as ensuring data integrity within a given project, a consistently implemented and properly documented normalization policy is essential for successful document interchange.

Character semantics

In addition to the Universal Character Set itself, the Unicode Consortium maintains a database of additional character semantics[9]. This includes names for each character code point and normative properties for it. Character properties, as given in this database, determine the semantics and thus the intended use of a code point or character. It also contains information that might be needed for correctly processing this character for different purposes. This database is an important reference in determining which Unicode code point to use to encode a certain character.

In addition to the printed documentation and lists made available by the Unicode consortium, the information it contains may also be accessed by a number of search systems over the Web (e.g. `http://www.eki.ee/letter/`). Examples of character properties included in the database include case, numeric value, directionality, and, where applicable status as a 'compatibility character'[10]. Where a project undertakes local definition of characters with code point in the PUA, it is desirable that any relevant additional information about the characters concerned should be recorded in an analogous way, as further discussed under *5. Representation of Non-standard Characters and Glyphs*.

vi.2.8 Character entities in non-validated documents

An important difference between SGML and XML is that the latter allows for the processing of non-validated documents. Since validity and validation are central TEI concerns, it is unlikely that documents prepared according to these Guidelines will ever be designed or implemented as merely well-formed in the XML sense. However in the domain of XML technologies, even where a document invokes a DTD or schema, it is not always necessarily the case that an XML processor will perform a full validation of it. XSLT transformation is a common case in point. By the workflow stage at which a document is handed off to an XSLT process for transformation, it is likely that its associated DTD or schema will already have fulfilled its role of integrity assurance and quality control, and so it may be undesirable to add validation to the processing overhead. For this reason, most XSLT processors do not attempt validation by default, even if a DTD or schema is declared and accessible. This can, however, create a problem where parsed entities, (and character entities in particular in the present context) are referenced. A validating parser reads all entity declarations from the DTD (including those for character entities) in the initial phase of processing, so that they can be resolved as and when required. However, where no validation takes place, it cannot automatically be assumed that the parser will be able to

[7] Available at `http://www.w3.org/TR/charmod`.

[8] available at `http://www.unicode.org/reports/tr15/`

[9] `http://www.unicode.org/ucd/`

[10] For further details, see *The Unicode Character Property Model* (Unicode Technical Report #23), at `http://www.unicode.org/reports/tr23/`.

resolve such entities in all circumstances. The XML standard requires a non-validating parser to read and act on entity declarations only if they are located within the document's internal subset (which does not, of course, mean that the entity declarations have to be manually merged into the document instance in advance of processing: character entity sets, for instance, count as being in the internal subset if they are placed there via a parameter entity, as is normal TEI practice). Some parsers when in non-validating mode will also access entity declarations in the external subset, but this behaviour is not mandated by the standard and should not be relied upon. Provided these facts are borne in mind, the presence of character entities in a document when parser validation is switched off should not cause any difficulties.

vi.2.9 Issues arising from the internal representations of Unicode

In theory it should not be necessary for encoders to have any knowledge of the various ways in which Unicode code points can be represented internally within a document or in the memory of a processing system, but experience shows that problems frequently arise in this area because of mistaken practice or defective software, and in order to recognise the resulting symptoms and correct their causes an outline knowledge of certain aspects of Unicode internal representation is desirable.

Encoding errors related to UTF-8

The code points assigned by Unicode 3.0 and later are notionally 32-bit integers, and the most straightforward way to represent each such integer in computer storage would be to use 4 eight-bit bytes. However, many of the code points for characters most commonly used in Latin scripts can be represented in one byte only and the vast majority of the remainder which are in common use (including those assigned from the most frequently used PUA range) can be expressed in two bytes alone. This accounts for the use of UTF-8 and UTF-16 and their special place in the XML standard. UTF-8 and UTF-16 are ways of representing 32-bit code points in an economical way.

UTF-8 is a variable length-encoding: the more significant bits there are in the underlying code point (or in everyday terminology the bigger the number used to represent the character), the more bytes UTF-8 uses to encode it. What makes UTF-8 particularly attractive for representing Latin scripts, explaining its status as the default encoding in XML documents, is that all code points that can be expressed in seven or fewer bits (the 127 values in the original ASCII character set) are also encoded as the same seven or fewer bits (and therefore in a single byte) in UTF-8. That is why a document which is actually encoded in pure 7-bit ASCII can be fed to an XML processor without alteration and without its encoding being explicitly declared: the processor will regard it as being in the UTF-8 representation of Unicode and be able to handle it correctly on that basis.

However, even within the domain of Latin-based scripts, some projects have documents which use characters from 8 bit extensions to ASCII, e.g. those in the ISO-8859-n series of encodings, and the way characters which under ISO-8859-n use all eight bits are encoded in UTF-8 is significantly different, giving rise to puzzling errors. Abstract characters that have a *single* byte code point where the highest bit is set (that is, they have a decimal numeric representation between 129 and 255) are encoded in ISO-8859-n as a *single*byte with the same value as the code point. But in UTF-8 code-point values inside that range are expressed as a *two* byte sequence. That is to say, the abstract character in question is no longer represented in the file or in memory by the same number as its code-point value: it is **transformed** (hence the T in UTF) into a sequence of two different numbers. Now as a side-effect of the way such UTF-8 sequences are derived from the underlying code-point value, many of the single-byte eight-bit values employed in ISO-8859-n encodings are illegal in UTF-8.

This complicated situation has a simple consequence which can cause great bewilderment. XML processors will effortlessly handle character data in pure 7-bit ASCII without that encoding needing to be declared to the parser, and will similarly accept documents encoded in an undeclared ISO-8859-n encoding if they happen to use no characters outside the strict ASCII subset of the ISO character sets; but the parse will immediately fail if an eight-bit character from an ISO-8859-n set is encountered in the input stream, unless the document's

encoding has been explicitly and correctly declared. Explicitly declaring the encoding ought to solve the problem, and if the file is correctly encoded throughout, it will do so. But since text editors and word processors are currently acquiring different degrees of Unicode support at different rates, projects are likely to find that they have to deal with some files encoded in UTF-8 along with others in, say, ISO-8859-1. Such encoding differences may go unnoticed, especially if the proportion of characters where the internal encodings are distinguishable is relatively small (for example in a long English text with a smattering of French words). If in the process of document preparation two such files have been merged, or intermixed via 'cut and paste' techniques, it is all too possible that the internal encodings of the resulting files will have become mixed as well. Thanks to misplaced notions of 'user friendliness' some current editing software silently corrects such miscodings as it displays the text, so that they remain hidden until the XML parser terminates with a fatal 'invalid character' error.

Where erroneously mixed encodings are the source of such an error, altering the encoding declaration will not solve the problem, though it may obfuscate it. Eight-bit character codes in a file declared as UTF-8 will always stop the parser. More insidiously, UTF-8 sequences in a file declared as ISO-8859-1 will not halt the parse, but will cause data corruption, because the parser will silently but erroneously convert each byte in every UTF-8 sequence into a spurious separate character, introducing semantic errors which may not become apparent until much later in the processing chain.

In projects that routinely handle documents in non-Latin scripts, everyone is well aware of the need to ensure correct and consistent encoding, so in such places mixed encoding problems seldom arise, and when they do are readily identified and remedied. Real confusion tends to arise, however, in projects which have a low awareness of the issues because they employ predominantly unaccented Latin characters, with only thinly-distributed instances of accented letters, or other 'special characters' where the internal representation under ISO-8859-n and UTF-8 are different (such as the copyright symbol, or, a frequent troublemaker where eventual HTML output is envisaged, the 'non-breaking space'). Even, or especially, if such projects view themselves as concerned only with English documents, the close relationship between XML and Unicode means they will need to acquire an understanding of these encoding issues and develop procedures which assure consistency and integrity of encoding and its correct declaration, including the use of appropriate software for transcoding and verification.

Encoding errors related to UTF-16

The advantages of UTF-8 as an internal representation of Unicode code points outlined above do not obtain where documents are in scripts other than Latin, Cyrillic or Hebrew. Where characters with code points in the sixteen-bit range (two-byte) predominate, UTF-8 is inappropriate, because it requires three or more bytes to represent each abstract character. Here the preferred representation of Unicode (which all XML-conformant parsers must support) is UTF-16, where each code point corresponding to an abstract character is represented in two eight-bit bytes[11]. This encoding presents a different hazard, especially while support for Unicode in editing software is relatively uneven and immature. Because the code points are represented as sixteen-bit integers stored (in most popular computers) in two separate bytes, the order in which those bytes are stored becomes important. This is dependent on the underlying hardware. In the realm of desktop computing, Macintosh machines, for example, store (on disk as well as in memory) byte pairs representing 16-bit integers with the higher-value byte first, whereas PCs using Intel processors store the bytes in the reverse order (this is often referred to with Swiftian nomenclature as big-endian versus little-endian byte order). This means that if a semantically identical plain text file encoded in UTF-16 is prepared on a Macintosh and on a PC, and the two files are then saved to disk, each byte pair in one file will be in the reverse order from the corresponding byte pair in the other file. To avoid the obvious incompatibility problems, the XML standard requires that all documents whose declared encoding is UTF-16 must begin with a special pseudo-character which is not itself part of the document, but merely a Byte Order Marker (BOM) from which the processor

[11] The use of 'surrogate' values to represent code points beyond the 16-bit range is passed over here, since it adds a complication that does not affect the key points at issue

can determine the byte order of the document that follows. Now the insertion of a correct BOM and the consistent maintenance of the byte order throughout the file ought to be taken care of transparently by software, but experience, especially from environments where work is distributed across big-endian and little-endian hardware, shows that this cannot always be taken for granted in the current state of software development. As with mixed encoding problems involving UTF-8, inconsistent byte-order in UTF-16 files seems to be the result of merging or cutting and pasting between files using software which does not correctly enforce byte order integrity, and out of misconceived 'user friendliness' which conceals byte-order inconsistencies from the user. Once more, the result can be files which look correct in an editor, but which the XML parser either rejects outright or silently passes on in a seriously garbled form. Again, to avoid the consequent errors, projects need to cultivate an informed awareness of relevant encoding issues and devise policies to avoid them in the first place or detect them at an early stage.

Chapter 1

The TEI Infrastructure

This chapter describes the infrastructure for the encoding scheme defined by these Guidelines. It introduces the conceptual framework within which the following chapters are to be understood, and the means by which that conceptual framework is implemented. It assumes some familiarity with XML and XML schemas (see chapter *v A Gentle Introduction to XML*) but is intended to be accessible to any user of these Guidelines. Other chapters supply further technical details, in particular chapter *22. Documentation Elements* which describes the XML schema used to express the Guidelines themselves, and chapter *23. Using the TEI* which combines a discussion of modification and conformance issues with a description of the intended behaviour of an ODD processor; these chapters should be read by anyone intending to implement a new TEI-based system.

The TEI encoding scheme consists of a number of modules, each of which declares particular XML elements and their attributes. Part of an element's declaration includes its assignment to one or more element classes. Another part defines its possible content and atttributes with reference to these classes. This indirection gives the TEI system much of its strength and its flexibility. Elements may be combined more or less freely to form a schema appropriate to a particular set of requirements. It is also easy to add new elements which reference existing classes or elements to a schema, as it is to exclude some of the elements provided by any module included in a schema.

In principle, a TEI schema may be constructed using any combination of modules. However, certain TEI modules are of particular importance, and should always be included in all but exceptional circumstances: the module tei described in the present chapter is of this kind because it defines classes, macros, and datatypes which are used by all other modules. The core module, defined in chapter *3. Elements Available in All TEI Documents* contains declarations for elements and attributes which are likely to be needed in almost any kind of document, and is therefore recommended for global use. The header module defined in chapter *2. The TEI Header* provides declarations for the metadata elements and attributes constituting the TEI Header, a component which is required for TEI conformance, while the textstructure module defined in chapter *4. Default Text Structure* declares basic structural elements needed for the encoding of most book-like objects. Most schemas will therefore need to include these four modules.

The specification for a TEI schema is itself a TEI document, using elements from the module described in chapter *22. Documentation Elements*: we refer to such a document informally as an ODD document, from the design goal originally formulated for the system: 'One Document Does it all'. Stylesheets for maintaining and processing ODD documents are maintained by the TEI, and these Guidelines are also maintained as such a document. As further discussed in *23.4. Implementation of an ODD System*, an ODD document can be processed to generate a schema expressed using any of the three schema languages currently in wide use: the XML DTD language, the ISO RELAX NG language, or the W3C Schema language, as well as to generate documentation such as the *Guidelines* and their associated web site.

The bulk of this chapter describes the TEI infrastructure module itself. Although it may be skipped at a first reading, an understanding of the topics addressed here is essential for anyone planning to take full advantage of the TEI customization techniques described in chapter *23.2. Personalization and Customization*.

The chapter begins by briefly characterizing each of the modules available in the TEI scheme. Section *1.2. Defining a TEI Schema* describes in general terms the method of constructing a TEI schema in a specific schema language such as XML DTD language, RELAX NG, or W3C Schema.

The next and largest part of the chapter introduces the attribute and element classes used to define groups of elements and their characteristics (section *1.3. The TEI Class System*).

Finally, section *1.4. Macros* introduces the concept of macros, which are used to express some commonly used content models, and lists the datatypes used to constrain the range of legal values for TEI attributes (section *1.4.2. Datatype Macros*).

1.1 TEI Modules

These Guidelines define several hundred elements and attributes for marking up documents of any kind. Each definition has the following components:

- a prose description
- a formal declaration, expressed using a special-purpose XML vocabulary defined by these Guidelines in combination with elements taken from the ISO schema language RELAX NG
- usage examples

Each chapter of the Guidelines presents a group of related elements, and also defines a corresponding set of declarations, which we call a module. All the definitions are collected together in the reference sections provided as an appendix. Formal declarations for a given chapter are collected together within the corresponding module. For convenience, each element is assigned to a single module, typically for use in some specific application area, or to support a particular kind of usage. A module is thus simply a convenient way of grouping together a number of associated element declarations. In the simple case, a TEI schema is made by combining together a small number of modules, as further described in section *1.2. Defining a TEI Schema* below.

The following table lists the modules defined by the current release of the Guidelines:

Module name	Formal public identifier	Where defined
analysis	Analysis and Interpretation	*17. Simple Analytic Mechanisms*
certainty	Certainty and Uncertainty	*21. Certainty and Responsibility*
core	Common Core	*3. Elements Available in All TEI Documents*
corpus	Metadata for Language Corpora	*15. Language Corpora*
dictionaries	Print Dictionaries	*9. Dictionaries*
drama	Performance Texts	*7. Performance Texts*
figures	Tables, Formulae, Figures	*14. Tables, Formulæ, and Graphics*
gaiji	Character and Glyph Documentation	*5. Representation of Non-standard Characters and Glyphs*
header	Common Metadata	*2. The TEI Header*
iso-fs	Feature Structures	*18. Feature Structures*
linking	Linking, Segmentation, and Alignment	*16. Linking, Segmentation, and Alignment*
msdescription	Manuscript Description	*10. Manuscript Description*
namesdates	Names, Dates, People, and Places	*13. Names, Dates, People, and Places*
nets	Graphs, Networks, and Trees	*19. Graphs, Networks, and Trees*

spoken	Transcribed Speech	*8. Transcriptions of Speech*
tagdocs	Documentation Elements	*22. Documentation Elements*
tei	TEI Infrastructure	*1. The TEI Infrastructure*
textcrit	Text Criticism	*12. Critical Apparatus*
textstructure	Default Text Structure	*4. Default Text Structure*
transcr	Transcription of Primary Sources	*11. Representation of Primary Sources*
verse	Verse	*6. Verse*

For each module listed above, the corresponding chapter gives a full description of the classes, elements, and macros which it makes available when it is included in a schema. Other chapters of these Guidelines explore other aspects of using the TEI scheme.

1.2 Defining a TEI Schema

To determine that an XML document is valid (as opposed to merely well-formed), its structure must be checked against a schema, as discussed in chapter *v A Gentle Introduction to XML*. For a valid TEI document, this schema must be a conformant TEI schema, as further defined in chapter *23.3. Conformance*. Local systems may allow their schema to be implicit, but for interchange purposes the schema associated with a document *must* be made explicit. The method of doing this recommended by these Guidelines is to provide explicitly or by reference a TEI schema specification against which the document may be validated.

A TEI-conformant schema is a specific combination of TEI modules, possibly also including additional declarations that modify the element and attribute declarations contained by each module, for example to suppress or rename some elements. The TEI provides an application-independent way of specifying a TEI schema by means of the <schemaSpec> element defined in chapter *22. Documentation Elements*. The same system may also be used to specify a schema which extends the TEI by adding new elements explicitly, or by reference to other XML vocabularies. In either case, the specification may be processed to generate a formal schema, expressed in a variety of specific schema languages, such as XML DTD language, RELAX NG, or W3C Schema. These output schemas can then be used by an XML processor such as a validator or editor to validate or otherwise process documents. Further information about the processing of a TEI formal specification is given in chapter *23. Using the TEI*.

1.2.1 A Simple Customization

The simplest customization of the TEI scheme combines just the four recommended modules mentioned above. In ODD format, this schema specification takes this form:

```
<schemaSpec ident="TEI-minimal" start="TEI">
  <moduleRef key="tei"/>
  <moduleRef key="header"/>
  <moduleRef key="core"/>
  <moduleRef key="textstructure"/>
</schemaSpec>
```

This schema specification contains references to each of four modules, identified by the **key** attribute on the <moduleRef> element. The schema specification itself is also given an identifier (TEI-minimal). An ODD processor will generate an appropriate schema from this set of declarations, expressed using the XML DTD language, the ISO RELAX NG language, the W3C Schema language, or in principle any other adequately powerful schema language. The resulting schema may then be associated with the document instance by one of a number of different mechanisms, as further described in chapter *v A Gentle Introduction to XML*. The start point (or root element) of document instances to be validated against the schema is specified by

means of the **start** attribute. Further information about the processing of an ODD specification is given in *23.4. Implementation of an ODD System*.

1.2.2 A Larger Customization

These Guidelines introduce each of the modules making up the TEI scheme one by one, and therefore, for clarity of exposition, each chapter focusses on elements drawn from a single module. In reality, of course, the markup of a text will draw on elements taken from many different modules, partly because texts are heterogenous objects, and partly because encoders have different goals. Some examples of this heterogeneity include:

- a text may be a collection of other texts of different types: for example, an anthology of prose, verse, and drama;
- a text may contain other smaller, embedded texts: for example, a poem or song included in a prose narrative;
- some sections of a text may be written in one form, and others in a different form: for example, a novel where some chapters are in prose, others take the form of dictionary entries, and still others the form of scenes in a play;
- an encoded text may include detailed analytic annotation, for example of rhetorical or linguistic features;
- an encoded text may combine a literal transcription with a diplomatic edition of the same or different sources;
- the description of a text may require additional specialised metadata elements, for example when describing manuscript material in detail.

The TEI provides mechanisms to support all of these and many other use cases. The architecture permits elements and attributes from any combination of modules to co-exist within a single schema. Within particular modules, elements and attributes are provided to support differing views of the 'granularity' of a text, for example:

- a definition of a corpus or collection as a series of <TEI> documents, sharing a common TEI header (see chapter *15. Language Corpora*)
- a definition of composite texts which combine optional front- and back-matter with a group of collected texts, themselves possibly composite (see section *4.3.1. Grouped Texts*)
- an element for the representation of embedded texts, where one narrative appears to 'float' within another (see section *4.3.2. Floating Texts*)

Subsequent chapters of these Guidelines describe in detail markup constructs appropriate for these and many other possible features of interest. The markup constructs can be combined as needed for any given set of applications or project.

For example, a project aiming to produce an ambitious digital edition of a collection of manuscript materials, to include detailed metadata about each source, digital images of the content, along with a detailed transcription of each source, and a supporting biographical and geographical database might need a schema combining several modules, as follows:

```
<schemaSpec ident="TEI-PROJECT" start="TEI">
  <moduleRef key="tei"/>
  <moduleRef key="header"/>
  <moduleRef key="core"/>
  <moduleRef key="textstructure"/>
  <moduleRef key="msdescription"/>
```

```
<!-- manuscript description -->
  <moduleRef key="transcr"/>
<!-- transcription of primary sources -->
  <moduleRef key="figures"/>
<!-- figures and tables -->
  <moduleRef key="namesdates"/>
<!-- names, dates, people, and places -->
</schemaSpec>
```

Alternatively, a simpler schema might be used for a part of such a project: those preparing the transcriptions, for example, might need only elements from the core, textstructure, and transcr modules, and might therefore prefer to use a simpler schema such as that generated by the following:

```
<schemaSpec ident="TEI-TRANSCR" start="TEI">
  <moduleRef key="tei"/>
  <moduleRef key="core"/>
  <moduleRef key="textstructure"/>
  <moduleRef key="transcr"/>
</schemaSpec>
```

The TEI architecture also supports more detailed customization beyond the simple selection of modules. A schema may suppress elements from a module, suppress some of their attributes, change their names, or even add new elements and attributes. Detailed discussion of the kind of modification possible in this way is provided in *23.2. Personalization and Customization* and conformance rules relating to their application are discussed in *23.3. Conformance*. These facilities are available for any schema language (though some features may not be available in all languages). The ODD language also makes it possible to combine TEI and non-TEI modules into a single schema, provided that the non-TEI module is expressed using the RELAX NG schema language (see further *22.6. Combining TEI and Non-TEI Modules*).

1.3 The TEI Class System

The TEI scheme distinguishes about five hundred different elements. To aid comprehension, modularity, and modification, the majority of these elements are formally classified in some way. Classes are used to express two distinct kinds of commonality among elements. The elements of a class may share some set of attributes, or they may appear in the same locations in a content model. A class is known as an attribute class if its members share attributes, and as a model class if its members appear in the same locations. In either case, an element is said to inherit properties from any classes of which it is a member.

Classes (and therefore elements which are members of those classes) may also inherit properties from other classes. For example, supposing that class A is a member (or a subclass) of class B, any element which is a member of class A will inherit not only the properties defined by class A, but also those defined by class B. In such a situation, we also say that class B is a superclass of class A. The properties of a superclass are inherited by all members of its subclasses.

A basic understanding of the classes into which the TEI scheme is organized is strongly recommended and is essential for any successful customization of the system.

1.3.1 Attribute Classes

An attribute class groups together elements which share some set of common attributes. Attribute classes are given names beginning `att.` and are usually adjectival. For example, the members of the class att.naming have in common a **key** and a **ref** attribute, both of which are inherited from their membership in the class rather than

individually defined for each element. These attribute is said to be defined by (or inherited from) the att.naming class. If another element were to be added to the TEI scheme for which these attributes were considered useful, the simplest way to provide them would be to make the new element a member of the att.naming class. Note also that this method ensures that the attributes in question are always defined in the same way, taking the same default values etc., no matter which element they are attached to.

Some attribute classes are defined within the tei infrastructural module and are thus globally available. Other attribute classes are specific to particular modules and thus defined in other chapters. Attributes defined by such classes will not be available unless the module concerned is included in a schema.

The attributes provided by an attribute class are those specified by the class itself, either directly, or by inheritance from another class. For example, the attribute class att.pointing.group provides attributes **domains** and **targFunc** to all of its members. This class is however a subclass of the att.pointing class, from which its members also inherit the attributes **type** and **evaluate**. Members of the class att.pointing will thus have these two attributes, while members of the class att.pointing.group will have all four.

Note that some modules define superclasses of an existing infrastructural class. For example, the global attribute class att.divLike makes attributes **org**, **part**, and **sample** available, while the att.metrical class, which is specific to the verse module, provides attributes **met**, **real**, and **rhyme**. Because att.metrical is defined as a superclass of att.divLike, all six of these attributes are available to elements; the declaration for att.metrical adds its three attributes to the three already defined by att.divLike when the verse module is included in a schema. If, however, this module is not included in a schema, then the att.divLike elements supplies only the three attributes first mentioned.

Attributes specific to particular modules are documented along with the relevant module rather than in the present chapter. One particular attribute class, known as att.global, is common to all modules, and is therefore described in some detail in the next section. A full list of all attribute classes is given in *Appendix B Attribute Classes* below.

1.3.1.1 Global Attributes

The following attributes are defined for every TEI element.

att.global provides attributes common to all elements in the TEI encoding scheme.

- @**xml:id** (identifier) provides a unique identifier for the element bearing the attribute.
- @**n** (number) gives a number (or other label) for an element, which is not necessarily unique within the document.
- @**xml:lang** (language) indicates the language of the element content using a 'tag' generated according to BCP 47
- @**rend** (rendition) indicates how the element in question was rendered or presented in the source text.
- @**rendition** points to a description of the rendering or presentation used for this element in the source text.
- @**xml:base** provides a base URI reference with which applications can resolve relative URI references into absolute URI references.

These attributes are optionally available for any TEI element; none of them is required.

1.3.1.1.1 Element Identifiers and Labels

The value supplied for the **xml:id** attribute must be a legal name, as defined in the World Wide Web Consortium's XML Recommendation. This means that it must begin with a letter, or the underscore character

('_'), and contain no characters other than letters, digits, hyphens, underscores, full stops, and certain combining and extension characters.[1]

In XML names (and thus the values of **xml:id** in an XML TEI document) uppercase and lowercase letters are distinguished, and thus *partTime* and *parttime* are two distinctly different names, and could (though perhaps unwisely) be used to denote two different element occurrences.

If two elements are given the same identifier, a validating XML parser will signal a syntax error. The following example, therefore, is *not* valid:

```
<p
xml:id="PAGE1"><q>What's it going to be then, eh?</q></p>
<p xml:id="PAGE1">There was me, that is Alex, and my three droogs,
that is Pete, Georgie, and Dim, ... </p>
```

Source: [23]

For a discussion of methods of providing unique identifiers for elements, see section *3.10.2. Creating New Reference Systems*.

The **n** attribute also provides an identifying name or number for an element, but in this case the information need not be a legal **xml:id** value. Its value may be any string of characters; typically it is a number or other similar enumerator or label. For example, the numbers given to the items of a numbered list may be recorded with the **n** attribute; this would make it possible to record errors in the numeration of the original, as in this list of chapters, transcribed from a faulty original in which the number 10 is used twice, and 11 is omitted:

```
<list type="ordered">
  <item n="1">About These Guidelines</item>
  <item n="2">A Gentle Introduction to SGML</item>
  <item n="9">Verse</item>
  <item n="10">Drama</item>
  <item n="10">Spoken Materials </item>
  <item n="12">Print Dictionaries</item>
</list>
```

The **n** attribute may also be used to record non-unique names associated with elements in a text, possibly together with a unique identifier as in the following example:

```
<div type="Book" n="One" xml:id="TXT0101">
<!-- ... -->
  <div type="stanza" n="xlii">
<!-- ... -->
  </div>
</div>
```

Source: [172]

As noted above there is no requirement to record a value for either the **xml:id** or the **n** attribute. Any XML processor can identify the sequential position of one element within another in an XML document without any additional tagging. An encoding in which each line of a long poem is explicitly labelled with its numerical sequence such as the following

[1] The colon is also by default a valid name character; however, it has a specific purpose in XML (to indicate namespace prefixes), and may not therefore be used in any other way within a name.

```
<l  n="1">
<!-- ... -->
</l>
<l  n="2">
<!-- ... -->
</l>
<l  n="3">
<!-- ... -->
</l>
<!-- ... -->
<l  n="100">
<!-- ... -->
</l>
```

is therefore probably redundant.

1.3.1.1.2 Language Indicators

The **xml:lang** attribute indicates the natural language and writing system applicable to the content of a given element. If it is not specified, the value is inherited from that of the immediately enclosing element. As a rule, therefore, it is simplest to specify the base language of the text on the <TEI> element, and allow most elements to take the default value for **xml:lang**; the language of an element then need be explicitly specified only for elements in languages other than the base language. It is strongly recommended that all language shifts in the source be explicitly identified by use of the **xml:lang** attribute, as further described in chapter *vi Languages and Character Sets*.

The values used for the **xml:lang** attribute must be constructed in a particular way, using values from standard lists. See further *vi.1 Language identification*.

The following two encodings convey the same information about the language of the text. In the first, the **xml:lang** attributes on the <emph> elements specify the same value as that on the parent <p> element, while in the second they inherit that value without specifying it.

```
<p  xml:lang="en"> ... Both parties deprecated war, but one of
them would <emph  xml:lang="en">make</emph> war rather than let
the nation survive, and the other would <emph  xml:lang="en">accept
  </emph> war rather than let it perish, and the war came.</p>
```

Source: [129]

```
<p  xml:lang="en"> ... Both parties deprecated war, but one of
them would <emph>make</emph> war rather than let
the nation survive, and the other would <emph>accept</emph>
war rather than let it perish, and the war came.</p>
```

In the following example, by contrast, the **xml:lang** attribute on the <term> element must be given if we wish to record the fact that the technical terms used are Latin rather than English; no **xml:lang** attribute is needed on the <q> element, by contrast, because it is in the same language as its parent.

```
<p  xml:lang="en">The constitution declares <q>that no bill of attainder
    or <term  xml:lang="la">ex post facto</term> law shall be passed.</q> ...</p>
```

Source: [138]

Note that additional information about a particular language may be supplied in the <language> element within the header (see section *2.4.2. Language Usage*).

1.3.1.1.3 Rendition Indicators

The **rend** attribute is used to give information about the physical presentation of the text in the source. In the following example, it is used to indicate that both the emphasized word and the proper name are printed in italics:

```
<p> ... Their motives <emph  rend="italics">might</emph> be
pure and pious; but he was equally alarmed by his knowledge
of the ambitious <name  rend="italics">Bohemond</name>, and
his ignorance of the Transalpine chiefs: ...</p>
```

Source: [84]

If all or most <emph> and <name> elements are rendered in the text by italics, it will be more convenient to register that fact in the TEI header once and for all (using the <rendition> element discussed below) and specify a **rend** value only for any elements which deviate from the stated rendition.

Although the contents of the **rend** attribute are free text, in any given project, encoders are advised to adopt a standard vocabulary with which to describe typographic or manuscript rendition of the text.

The <rendition> element defined in *2.3.4. The Tagging Declaration* may be used to hold such descriptions, expressed in free text, or using a formal language. A <rendition> element can then be associated with any element, either by default, or by means of the global **rendition** attribute. For example:

```
<!-- define italic style using CSS --><rendition  xml:id="IT"  scheme="css">font-style: italic</rendition>
<!-- set italic style as default for the emph and hi elements -->
<tagUsage  gi="emph"  render="#IT"/>
<tagUsage  gi="hi"  render="#IT"/>
<!-- indicate that a specific p element is also in italic style -->
<p  rendition="#IT"/>
```

The **rendition** attribute always points to one or more <rendition> elements, each of which defines some aspect of the rendering or appearance of the text in its original form. These details may be described using a formal language, such as CSS (Lie and Bos (eds.) (1999)) or XSL-FO (Berglund (ed.) (2006)); in some other formal language developed for a specific project; or informally in running prose. Although languages such as CSS and XSL-FO are generally used to describe document output to screen or print, they nonetheless provide formal and precise mechanisms for describing the appearance of many source documents, especially print documents, but also many aspects of manuscript documents. For example, both CSS and XSL-FO provide mechanisms for describing typefaces, weight, and styles; character and line spacing; and so on.

If both **rendition** and **rend** attributes are provided for a given element, the latter always takes precedence. The **rendition** attribute is analogous to the X/HTML **class** attribute, which references style declarations in a Cascading Style Sheet. The **rend** attribute is analogous to the XHTML or HTML **style** attribute, which provides a mechanism for embedding inline rendition information at the point of use within a document. Note that, in either case, the TEI attributes describe the rendition or appearance of the source document, *not* intended output renditions, although often the two may be closely related.

1.3.2 Model Classes

As noted above, the members of a given TEI model class share the property that they can all appear in the same location within a document. Wherever possible, the content model of a TEI element is expressed not directly in terms of specific elements, but indirectly in terms of particular model classes. This makes content models simpler and more consistent; it also makes them much easier to understand and to modify.

Like attribute classes, model classes may have subclasses or superclasses. Just as elements inherit from a class the ability to appear in certain locations of a document (wherever the class can appear), so all members of a subclass inherit the ability to appear wherever any superclass can appear. To some extent, the class system thus provides a way of reducing the whole TEI galaxy of elements into a tidy hierarchy. This is however not entirely the case.

In fact, the nature of a given class of elements can be considered along two dimensions: as noted, it defines a set of places where the class members are permitted within the document hierarchy; it also implies a semantic grouping of some kind. For example, the very large class of elements which can appear within a paragraph comprises a number of other classes, all of which have the same structural property, but which differ in their field of application. Some are related to highlighting, while others relate to names or places, and so on. In some cases, the 'set of places where class members are permitted' is very constrained: it may just be within one specific element, or one class of element, for example. In other cases, elements may be permitted to appear in very many places, or in more than one such set of places.

These factors are reflected in the way that model classes are named. If a model class has a name containing **part**, such as model.divPart or model.biblPart then it is primarily defined in terms of its structural location. For example, those elements (or classes of element) which appear as content of a constitute the model.divPart class; those which appear as content of a <bibl> constitute the model.biblPart class. If, however, a model class has a name containing **like**, such as model.biblLike or model.nameLike, the implication is that its members all have some additional semantic property in common, for example containing a bibliographic description, or containing some form of name, respectively. These semantically-motivated classes often provide a useful way of dividing up large structurally-motivated classes: for example, the very general structural class model.pPart.data ('data elements that form part of a paragraph') has four semantically-motivated member classes (model.addressLike, model.dateLike, model.measureLike, and model.nameLike), the last of these being itself a superclass with three members.

Although most classes are defined by the tei infrastructure module, a class cannot be populated unless some other specific module is included in a schema, since element declarations are contained by modules. Classes are not declared 'top down', but instead gain their members as a consequence of individual elements' declaration of their membership. The same class may therefore contain different members, depending on which modules are active. Consequently, the content model of a given element (being expressed in terms of model classes) may differ depending on which modules are active.

Some classes contain only a single member, even when all modules are loaded. One reason for declaring such a class is to make it easier for a customization to add new member elements in a specific place, particularly in areas where the TEI does not make fully elaborated proposals. For example, the TEI class model.rdgLike, initially empty, is expanded by the textcrit module to include just the TEI <rdg> element. A project wishing to add an alternative way of structuring text-critical information could do so by defining their own elements and adding it to this class.

Another reason for declaring single-member classes is where the class members are not needed in all documents, but appear in the same place as elements which are very frequently required. For example, the specialised element <g> used to represent a non-Unicode character or glyph is provided as the only member of the model.gLike class when the gaiji module is added to a schema. References to this class are included in almost every content model, since if it is used at all the <g> must be available wherever text is available; however these references have no effect unless the gaiji module is loaded.

At the other end of the scale, a few of the classes predefined by the tei module are subsequently populated with very many members. For example, the class model.pPart groups all the classes of element which can appear within a <p> or paragraph element. The core module alone adds more than fifty elements to this class; the namesdates module adds another twenty, as does the tagdocs module. Since the <p> element is one of the basic building blocks of a TEI document it is not surprising that each module will need to add elements to it. The class system here provides a very convenient way of controlling the resulting complexity. Typically,

elements are not added directly to these very general classes, but via some intermediate semantically-motivated class.

Just as there are a few classes which have a single member, so there are some classes which are used only once in the TEI architecture. These classes, which have no superclass and therefore do not fit into the class hierarchy defined here, are a convenient way of maintaining elements which are highly structured internally, but which appear from the outside to be uniform objects like others at the same level.[2] Members of such classes can only ever appear within one element, or one class of elements. For example, the class model.addrPart is used only to express the content model for the element <address>; it references some other classes of elements, which can appear elsewhere, and also some elements which can only appear inside an address.

1.3.2.1 Basic Model Classes

The TEI class system makes the following threefold division of elements:

divisions high level, possibly self-nesting, major divisions of texts. These elements populate the classes model.divLike, model.div1Like, etc.

chunks elements such as paragraphs and other paragraph-level elements, which can appear directly within texts or within such divisions, but not within other chunks. These elements populate the class model.divPart, either directly or by means of other classes such as model.pLike (paragraph-like elements), model.entryLike, etc.

phrase-level elements elements such as highlighted phrases, book titles, or editorial corrections which can occur only within chunks (paragraphs or paragraph-level elements), but not between them (and thus cannot appear directly within a division). These elements populate the class model.phrase.[3]

The TEI identifies the following fundamental groupings derived from these three:

inter-level elements elements such as lists, notes, quotations, etc. which can appear either between chunks (as children of a) or within them; these elements populate the class model.inter. Note that this class is not a superset of the model.phrase and model.chunk classes but rather the group of elements which are both chunk-like and phrase-like; the classes model.phrase, model.pLike, and model.inter are all disjoint.

components elements which can appear directly within texts or text divisions; this is a combination of the inter- and chunk- level elements defined above. These elements populate the class model.common, which is defined as a superset of the classes model.divPart, model.inter, and (when the dictionary module is included in a schema) model.entryLike.

Broadly speaking, the front, body, and back of a text each comprises a series of components, optionally grouped into divisions.

As noted above, some elements and element classes belong to none of these groupings; however, over two-thirds of the 500+ elements defined in the present edition of these Guidelines are classified in this way. Future editions of these recommendations will extend and develop this classification scheme.

A complete alphabetical list of all model classes is provided in *Appendix A Model Classes*.

[2] In former editions of these Guidelines, such elements were known metaphorically as 'crystals'.

[3] Note that in this context, phrase means any string of characters, and can apply to individual words, parts of words, and groups of words indifferently; it does not refer only to linguistically-motivated phrasal units. This may cause confusion for readers accustomed to applying the word in a more restrictive sense.

Content model	Number of elements using this	Description
macro.phraseSeq	83	any combination of text with elements from the model.gLike, model.global, or model.phrase classes
macro.paraContent	49	macro.phraseSeq with the addition of model.inter
empty	39	elements that have no content
macro.specialPara	24	macro.paraContent with the addition of model.divPart
macro.phraseSeq.limited	24	a subset of model.phraseSeq appropriate for use in non-transcriptional contexts
text	21	plain untagged text
macro.xtext	19	any combination of text with elements from the model.gLike class

Table 1.2:

1.4 Macros

The infrastructure module defined by this chapter also declares a number of macros, or shortcut names for frequently occurring parts of other declarations. Macros are used in two ways in the TEI scheme: to stand for frequently-encountered content models, or parts of content models (*1.4.1. Standard Content Models*); and to stand for attribute datatypes (*1.4.2. Datatype Macros*).

1.4.1 Standard Content Models

As far as possible, the TEI schemas use the following set of frequently-encountered content models to help achieve consistency among different elements.

macro.paraContent (paragraph content) defines the content of paragraphs and similar elements.

macro.limitedContent (paragraph content) defines the content of prose elements that are not used for transcription of extant materials.

macro.phraseSeq (phrase sequence) defines a sequence of character data and phrase-level elements.

macro.phraseSeq.limited (limited phrase sequence) defines a sequence of character data and those phrase-level elements that are not typically used for transcribing extant documents.

macro.schemaPattern provides a pattern to match elements from the chosen schema language

macro.specialPara ('special' paragraph content) defines the content model of elements such as notes or list items, which either contain a series of component-level elements or else have the same structure as a paragraph, containing a series of phrase-level and inter-level elements.

macro.xtext (extended text) defines a sequence of character data and gaiji elements.

The present version of the TEI Guidelines includes some 500 different elements.Table 1 shows, in descending order of frequency, the seven most commonly used content models.

1.4.2 Datatype Macros

The values which attributes may take in a TEI schema are defined, for the most part, by reference to a TEI datatype. Each such datatype is defined in terms of other primitive datatypes, derived mostly from W3C Schema Datatypes, literal values, or other datatypes. This indirection makes it possible for a TEI application to set constraints either globally or in individual cases, by redefining the datatype definition or the reference to it respectively. In some cases, the TEI datatype includes additional usage constraints which cannot be enforced by existing schema languages, although a TEI-compliant processor should attempt to validate them (see further discussion in chapter *23.3. Conformance*).

Where literal values or name tokens are used in a datatype definition, an associated value list supplies definitions for the significance of suggested or (in the case of closed lists) all possible values.

TEI-defined datatypes may be grouped into those which define normalised values for numeric quantities, probabilities, or temporal expressions, those which define various kinds of shorthand codes or keys, and those which define pointers or links.

The following datatypes are used for attributes which are intended to hold normalized values of various kinds. First, expressions of quantity or probability:

data.certainty defines the range of attribute values expressing a degree of certainty.

data.probability defines the range of attribute values expressing a probability.

data.numeric defines the range of attribute values used for numeric values.

data.count defines the range of attribute values used for a non-negative integer value used as a count.

Examples of attributes using the data.probability datatype include **degree** on <damage> or <certainty>; examples of data.numeric include **quantity** on members of the att.measurement class or **value** on <numeric>; examples of data.count include **cols** on <cell> and <table>.

Next, the datatypes used for attributes which are intended to hold normalized dates or times, durations, or truth values:

data.duration.w3c defines the range of attribute values available for representation of a duration in time using W3C datatypes.

data.duration.iso defines the range of attribute values available for representation of a duration in time using ISO 8601 standard formats

data.temporal.w3c defines the range of attribute values expressing a temporal expression such as a date, a time, or a combination of them, that conform to the W3C *XML Schema Part 2: Datatypes* specification.

data.temporal.iso defines the range of attribute values expressing a temporal expression such as a date, a time, or a combination of them, that conform to the international standard *Data elements and interchange formats – Information interchange – Representation of dates and times.*

data.truthValue defines the range of attribute values used to express a truth value.

data.xTruthValue (extended truth value) defines the range of attribute values used to express a truth value which may be unknown.

data.language defines the range of attribute values used to identify a particular combination of human language and writing system.

data.sex defines the range of attribute values used to identify human or animal sex.

Note that in each of these cases the values used are those recommended by existing international standards: ISO 8601 as profiled by *XML Schema Part 2: Datatypes Second Edition* in the case of durations, times, and date; W3C Schema datatypes in the case of truth values; BCP 47 in the case of language; and ISO 5218 in the case of sex.

The following datatypes have more specialised uses:

data.outputMeasurement defines a range of values for use in specifying the size of an object that is intended for display on the web.

data.namespace defines the range of attribute values used to indicate XML namespaces as defined by the W3C Namespaces in XML Technical Recommendation.

data.pattern (regular expression pattern) defines attribute values which are expressed as a regular expression.

data.pointer defines the range of attribute values used to provide a single URI pointer to any other resource, either within the current document or elsewhere.

By far the largest number of TEI attributes take values which are coded values or names of some kind. These values may be constrained or defined in a number of different ways, each of which is given a different name, as follows:

data.key defines the range of attribute values expressing a coded value by means of an arbitrary identifier, typically taken from a set of externally-defined possibilities.

data.word defines the range of attribute values expressed as a single word or token.

data.name defines the range of attribute values expressed as an XML Name.

data.enumerated defines the range of attribute values expressed as a single XML name taken from a list of documented possibilities.

data.code defines the range of attribute values expressing a coded value by means of a pointer to some other element which contains a definition for it.

The attribute **key** provided by the att.naming class is currently the only attribute of type data.key. It is used to supply an externally-defined identifier, such as a database key or filename. Because such identifiers are externally-defined, no constraints are placed on their possible values: any string of Unicode characters may be used. Any constraints on their values, such as the rules for constructing a valid database key in a particular system, may be documented by a <tagUsage> element in the TEI Header, but are not enforced by the datatype as defined here. Such system-specific constraints may however be added to a TEI schema by using the customisation techniques methods described in *23.2. Personalization and Customization*.

Attributes of type data.word, such as **age** on <person>, are used to supply an identifier expressed as any kind of single token or word. The TEI places a few constraints on the characters which may be used for this purpose: only Unicode characters classified as letters, digits, punctuation characters, or symbols can appear in an attribute value of this kind. Note in particular that such values cannot include whitespace characters. Legal values include **cholmondeley**, **été**, **1234**, **_content**, or **xml:id**, but not **grand wazoo**. Attributes of this kind are sometimes used to associate (by co-reference) elements of different types.

Attributes of type data.name are also words in this sense, but they have the additional constraint that they must be legal XML identifiers, as defined by the XML 1.0 specification, or successors. As such, they may not begin with digits or punctuation characters. Legal identifiers include **cholmondeley**, **été**, **e_content**, or **xml:id**, but not **grand wazoo** or **1234**. Attributes of this kind are typically used to represent XML element or attribute names.

Attributes of type data.enumerated, such as **new** on <shift> or **evidence** supplied by att.editLike, have the same definition as data.word above, with the added constraint that the word supplied is taken from a specific list of possibilities. In each case, the element or class specification which includes the definition for the attribute will also contain a list of possible values, together with a prose description of their intended significance. This list may be open (in which case the list is advisory), or closed (in which case it determines the range of legal values). In this latter case, the datatype will not be data.enumerated, but an explicit list of the possible values.

Attributes of type data.code are similar in function, in that they also supply encoded names for values which are defined in more detail elsewhere. In this case, however, the full definition is supplied as content of another XML element, typically but not necessarily in the same document, and it is referenced by means of a pointer.

An attribute may, of course, take more than one value of a given type, for example a list of pointer values, or a list of words. In the TEI scheme, this information is regarded as a property of the <datatype> element used to document the attribute in question rather than as a distinct 'datatype'. See further *22.4.5.1. Datatypes*.

1.5 The TEI Infrastructure Module

The tei module defined by this chapter is a required component of any TEI schema. It provides declarations for all datatypes, and initial declarations for the attribute classes, model classes, and macros used by other modules in the TEI scheme. Its components are listed below in alphabetical order:

Module tei: Declarations for classes, datatypes, and macros available to all TEI modules

- Classes defined: att.ascribed att.damaged att.datable att.datable.w3c att.declarable att.declaring att.dimensions att.divLike att.duration.w3c att.editLike att.global att.handFeatures att.internetMedia att.interpLike att.measurement att.naming att.personal att.placement att.segLike att.spanning att.tableDecoration att.timed att.transcriptional att.translatable att.typed att.xmlspace model.addrPart model.addressLike model.biblLike model.biblPart model.castItemPart model.catDescPart model.choicePart model.common model.dateLike model.div1Like model.div2Like model.div3Like model.div4Like model.div5Like model.div6Like model.div7Like model.divBottom model.divBottomPart model.divGenLike model.divLike model.divPart model.divTop model.divTopPart model.divWrapper model.egLike model.emphLike model.entryPart model.entryPart.top model.featureVal model.featureVal.complex model.featureVal.single model.frontPart model.frontPart.drama model.gLike model.global model.global.edit model.global.meta model.glossLike model.graphicLike «model.handDescPart» model.headLike model.hiLike model.highlighted model.imprintPart model.inter model.lLike model.lPart model.labelLike model.limitedPhrase model.listLike model.measureLike model.milestoneLike model.msItemPart model.nameLike model.nameLike.agent model.noteLike model.oddDecl model.oddRef model.offsetLike model.orgStateLike model.pLike model.pLike.front model.pPart.data model.pPart.edit model.pPart.editorial model.pPart.msdesc model.pPart.transcriptional model.persEventLike model.persStateLike model.persTraitLike model.personLike model.personPart model.phrase model.phrase.xml model.physDescPart model.placeEventLike model.placeLike model.placeNamePart model.placeStateLike model.placeTraitLike model.ptrLike model.publicationStmtPart model.qLike model.quoteLike model.resourceLike model.respLike model.segLike model.settingPart model.specDescLike model.stageLike model.textDescPart model.titlepagePart
- Macros defined: data.certainty data.code data.count data.duration.iso data.duration.w3c data.enumerated data.key data.language data.name data.namespace data.numeric data.outputMeasurement data.pattern data.pointer data.probability data.sex data.temporal.iso data.temporal.w3c data.truthValue data.word data.xTruthValue macro.limitedContent macro.paraContent macro.phraseSeq macro.phraseSeq.limited macro.specialPara macro.xtext

The order in which declarations are made within the infrastructure module is critical, since several class declarations refer to others, which must therefore precede them. Other constraints on the order of declarations derive from the way in which the modularity of the TEI scheme is implemented in different schema languages. The XML DTD fragment implementing this TEI module makes extensive use of parameter entities and marked sections to effect a kind of conditional construction; the RELAX NG schema fragment similarly predeclares a number of patterns with null ('notAllowed') values. These issues are further discussed in chapter *23.4. Implementation of an ODD System*.

Chapter 2

The TEI Header

This chapter addresses the problems of describing an encoded work so that the text itself, its source, its encoding, and its revisions are all thoroughly documented. Such documentation is equally necessary for scholars using the texts, for software processing them, and for cataloguers in libraries and archives. Together these descriptions and declarations provide an electronic analogue to the title page attached to a printed work. They also constitute an equivalent for the content of the code books or introductory manuals customarily accompanying electronic data sets.

Every TEI-conformant text must carry such a set of descriptions, prefixed to it and encoded as described in this chapter. The set is known as the TEI header, tagged <teiHeader>, and has four major parts:

1. a file description, tagged <fileDesc>, containing a full bibliographical description of the computer file itself, from which a user of the text could derive a proper bibliographic citation, or which a librarian or archivist could use in creating a catalogue entry recording its presence within a library or archive. The term computer file here is to be understood as referring to the whole entity or document described by the header, even when this is stored in several distinct operating system files. The file description also includes information about the source or sources from which the electronic document was derived. The TEI elements used to encode the file description are described in section *2.2. The File Description* below.

2. an encoding description, tagged <encodingDesc>, which describes the relationship between an electronic text and its source or sources. It allows for detailed description of whether (or how) the text was normalized during transcription, how the encoder resolved ambiguities in the source, what levels of encoding or analysis were applied, and similar matters. The TEI elements used to encode the encoding description are described in section *2.3. The Encoding Description* below.

3. a text profile, tagged <profileDesc>, containing classificatory and contextual information about the text, such as its subject matter, the situation in which it was produced, the individuals described by or participating in producing it, and so forth. Such a text profile is of particular use in highly structured composite texts such as corpora or language collections, where it is often highly desirable to enforce a controlled descriptive vocabulary or to perform retrievals from a body of text in terms of text type or origin. The text profile may however be of use in any form of automatic text processing. The TEI elements used to encode the profile description are described in section *2.4. The Profile Description* below.

4. a revision history, tagged <revisionDesc>, which allows the encoder to provide a history of changes made during the development of the electronic text. The revision history is important for version control and for resolving questions about the history of a file. The TEI elements used to encode the revision description are described in section *2.5. The Revision Description* below.

A TEI header can be a very large and complex object, or it may be a very simple one. Some application areas (for example, the construction of language corpora and the transcription of spoken texts) may require more specialized and detailed information than others. The present proposals therefore define both a core set of elements (all of which may be used without formality in any TEI header) and some additional elements which become available within the header as the result of including additional specialized modules within the schema. When the module for language corpora (described in chapter *15. Language Corpora*) is in use, for example, several additional elements are available, as further detailed in that chapter.

The next section of the present chapter briefly introduces the overall structure of the header and the kinds of data it may contain. This is followed by a detailed description of all the constituent elements which may be used in the core header. Section *2.6. Minimal and Recommended Headers* , at the end of the present chapter, discusses the recommended content of a minimal TEI header and its relation to standard library cataloguing practices.

2.1 Organization of the TEI Header

2.1.1 The TEI Header and its Components

The <teiHeader> element should be clearly distinguished from the front matter of the text itself (for which see section *4.5. Front Matter*). A composite text, such as a corpus or collection, may contain several headers, as further discussed below. In the usual case, however, a TEI-conformant text will contain a single <teiHeader> element, followed by a single <text> element.

The header element has the following description:

<teiHeader> (TEI Header) supplies the descriptive and declarative information making up an electronic title page prefixed to every TEI-conformant text.

@**type** specifies the kind of document to which the header is attached, for example whether it is a corpus or individual text.

As discussed above, the <teiHeader> element has four principal components:

<fileDesc> (file description) contains a full bibliographic description of an electronic file.

<encodingDesc> (encoding description) documents the relationship between an electronic text and the source or sources from which it was derived.

<profileDesc> (text-profile description) provides a detailed description of non-bibliographic aspects of a text, specifically the languages and sublanguages used, the situation in which it was produced, the participants and their setting.

<revisionDesc> (revision description) summarizes the revision history for a file.

Of these, only the <fileDesc> element is required in all TEI headers; the others are optional. The top level elements in the full form of a TEI header are thus:

```
<teiHeader>
  <fileDesc>
<!-- ... -->
  </fileDesc>
  <encodingDesc>
<!-- ... -->
  </encodingDesc>
  <profileDesc>
<!-- ... -->
  </profileDesc>
  <revisionDesc>
<!-- ... -->
```

```
  </revisionDesc>
</teiHeader>
```

while a minimal header takes the form:

```
<teiHeader>
  <fileDesc>
<!-- ... -->
  </fileDesc>
</teiHeader>
```

In the case of language corpora or collections, it may be desirable to record header information either at the level of the individual components in the corpus or collection, or at the level of the corpus or collection itself (more details concerning the tagging of composite texts are given in section *15. Language Corpora*, which should be read in conjunction with the current chapter). The **type** attribute may be used to indicate whether the header applies to a corpus or a single text. A corpus may thus take the form:

```
<teiCorpus>
  <teiHeader type="corpus">
<!-- corpus-level metadata here -->
  </teiHeader>
  <TEI>
    <teiHeader type="text">
<!-- metadata specific to this text here -->
    </teiHeader>
    <text>
<!-- ... -->
    </text>
  </TEI>
  <TEI>
    <teiHeader type="text">
<!-- metadata specific to this text here -->
    </teiHeader>
    <text>
<!-- ... -->
    </text>
  </TEI>
</teiCorpus>
```

2.1.2 Types of Content in the TEI Header

The elements occurring within the TEI header may contain several types of content; the following list indicates how these types of content are described in the following sections:

free prose Most elements contain simple running prose at some level. Many elements may contain either prose (possibly organized into paragraphs) or more specific elements, which themselves contain prose. In this chapter's descriptions of element content, the phrase *prose description* should be understood to imply a series of paragraphs, each marked as a <p> element. The word *phrase*, by contrast, should be understood to imply character data, interspersed as need be with phrase-level elements, but not organized into paragraphs. For more information on paragraphs, highlighted phrases, lists, etc., see section *3.1. Paragraphs*.

grouping elements Elements whose names end with the suffix *Stmt* (e.g. <editionStmt>, <titleStmt>) usually enclose a group of specialized elements recording some structured information. In the case of the bibliographic elements, the suffix *Stmt* is used in names of elements corresponding to the 'areas' of the International Standard Bibliographic Description.[1] In most cases grouping elements may contain prose descriptions as an alternative to the set of specialized elements, thus allowing the encoder to choose whether or not the information concerned should be presented in a structured form or in prose.

declarations Elements whose names end with the suffix *Decl* (e.g. <tagsDecl>, <refsDecl>) enclose information about specific encoding practices applied in the electronic text; often these practices are described in coded form. Typically, such information takes the form of a series of declarations, identifying a code with some more complex structure or description. A declaration which applies to more than one text or division of a text need not be repeated in the header of each such text or subdivision. Instead, the **decls** attribute of each text (or subdivision of the text) to which the declaration applies may be used to supply a cross-reference to it, as further described in section *15.3. Associating Contextual Information with a Text*.

descriptions Elements whose names end with the suffix *Desc* (e.g. <settingDesc>, <projectDesc>) contain a prose description, possibly, but not necessarily, organized under some specific headings by suggested sub-elements.

2.1.3 Model Classes in the TEI Header

The TEI Header provides a very rich collection of metadata categories, but makes no claim to be exhaustive. It is certainly the case that individual projects may wish to record specialised metadata which either does not fit within one of the predefined categories identified by the TEI Header or requires a more specialized element structure than is proposed here. To overcome this problem, the encoder may elect to define additional elements using the customization methods discussed in *23.2. Personalization and Customization*. The TEI class system makes such customizations simpler to effect and easier to use in interchange.

These classes are specific to parts of the header:

model.applicationLike groups elements used to record application-specific information about a document in its header.

model.catDescPart groups component elements of the TEI Header Category Description.

model.editorialDeclPart groups elements which may be used inside <editorialDecl> and appear multiple times.

model.encodingPart groups elements which may be used inside <encodingDesc> and appear multiple times.

model.profileDescPart groups elements which may be used inside <profileDesc> and appear multiple times.

model.headerPart groups high level elements which may appear more than once in a TEI Header.

model.sourceDescPart groups elements which may be used inside <sourceDesc> and appear multiple times.

model.textDescPart groups elements used to categorise a text for example in terms of its situational parameters.

[1] For more information on this highly influential family of standards, first proposed in 1969 by the International Federation of Library Associations, see `http://www.ifla.org/VII/s13/pubs/isbd.htm`. On the relation between the TEI proposals and other standards for bibliographic description, see further section *2.7. Note for Library Cataloguers*.

2.2 The File Description

This section describes the <fileDesc> element, which is the first component of the <teiHeader> element.

The bibliographic description of a machine-readable or digital text resembles in structure that of a book, an article, or any other kind of textual object. The file description element of the TEI header has therefore been closely modelled on existing standards in library cataloguing; it should thus provide enough information to allow users to give standard bibliographic references to the electronic text, and to allow cataloguers to catalogue it. Bibliographic citations occurring elsewhere in the header, and also in the text itself, are derived from the same model (on bibliographic citations in general, see further section *3.11. Bibliographic Citations and References*). See further section *2.7. Note for Library Cataloguers*.

The bibliographic description of an electronic text should be supplied by the mandatory <fileDesc> element:

<fileDesc> (file description) contains a full bibliographic description of an electronic file.

The <fileDesc> element contains three mandatory elements and four optional elements, each of which is described in more detail in sections *2.2.1. The Title Statement* to *2.2.6. The Notes Statement* below. These elements are listed below in the order in which they must be given within the <fileDesc> element.

<titleStmt> (title statement) groups information about the title of a work and those responsible for its intellectual content.

<editionStmt> (edition statement) groups information relating to one edition of a text.

<extent> describes the approximate size of a text as stored on some carrier medium, whether digital or non-digital, specified in any convenient units.

<publicationStmt> (publication statement) groups information concerning the publication or distribution of an electronic or other text.

<seriesStmt> (series statement) groups information about the series, if any, to which a publication belongs.

<notesStmt> (notes statement) collects together any notes providing information about a text additional to that recorded in other parts of the bibliographic description.

<sourceDesc> (source description) supplies a description of the source text(s) from which an electronic text was derived or generated.

A file description containing all possible sub-elements has the following structure:

```
<teiHeader>
  <fileDesc>
    <titleStmt>
<!-- ... -->
    </titleStmt>
    <editionStmt>
<!-- ... -->
    </editionStmt>
    <extent>
<!-- ... -->
    </extent>
    <publicationStmt>
<!-- ... -->
    </publicationStmt>
    <seriesStmt>
<!-- ... -->
    </seriesStmt>
    <notesStmt>
<!-- ... -->
```

```
   </notesStmt>
   <sourceDesc>
<!-- ... -->
   </sourceDesc>
  </fileDesc>
</teiHeader>
```

Several of these elements may be omitted; a minimal file description has the following structure:

```
<teiHeader>
  <fileDesc>
   <titleStmt>
<!-- ... -->
   </titleStmt>
   <publicationStmt>
<!-- ... -->
   </publicationStmt>
   <sourceDesc>
<!-- ... -->
   </sourceDesc>
  </fileDesc>
<!-- other optional parts of the header here -->
</teiHeader>
```

2.2.1 The Title Statement

The <titleStmt> element is the first component of the <fileDesc> element, and is mandatory:

<titleStmt> (title statement) groups information about the title of a work and those responsible for its intellectual content.

It contains the title given to the electronic work, together with one or more optional statements of responsibility which identify the encoder, author, compiler, or other parties responsible for it:

<title> contains the full title of a work of any kind.

<author> in a bibliographic reference, contains the name of the author(s), personal or corporate, of a work; the primary statement of responsibility for any bibliographic item.

<sponsor> specifies the name of a sponsoring organization or institution.

<funder> (funding body) specifies the name of an individual, institution, or organization responsible for the funding of a project or text.

<principal> (principal researcher) supplies the name of the principal researcher responsible for the creation of an electronic text.

<respStmt> (statement of responsibility) supplies a statement of responsibility for the intellectual content of a text, edition, recording, or series, where the specialized elements for authors, editors, etc. do not suffice or do not apply.

<resp> (responsibility) contains a phrase describing the nature of a person's intellectual responsibility.

<name> (name, proper noun) contains a proper noun or noun phrase.

The <title> element contains the chief name of the electronic work, including any alternative title or subtitles it may have. It may be repeated, if the work has more than one title (perhaps in different languages) and takes whatever form is considered appropriate by its creator. Where the electronic work is derived from an existing source text, it is strongly recommended that the title for the former should be derived from the latter, but clearly distinguishable from it, for example by the addition of a phrase such as ': an electronic transcription'

or 'a digital edition'. This will distinguish the electronic work from the source text in citations and in catalogues which contain descriptions of both types of material.

The electronic work will also have an external name (its 'filename' or 'data set name') or reference number on the computer system where it resides at any time. This name is likely to change frequently, as new copies of the file are made on the computer system. Its form is entirely dependent on the particular computer system in use and thus cannot always easily be transferred from one system to another. Moreover, a given work may be composed of many files. For these reasons, these Guidelines strongly recommend that such names should *not* be used as the <title> for any electronic work.

Helpful guidance on the formulation of useful descriptive titles in difficult cases may be found in the Anglo-American Cataloguing Rules (Gorman and Winkler, 1978, chapter 25) or in equivalent national-level bibliographical documentation.

The elements <author>, <sponsor>, <funder>, and <principal>, are specializations of the more general <respStmt> element. These elements are used to provide the statements of responsibility which identify the person(s) responsible for the intellectual or artistic content of an item and any corporate bodies from which it emanates.

Any number of such statements may occur within the title statement. At a minimum, identify the author of the text and (where appropriate) the creator of the file. If the bibliographic description is for a corpus, identify the creator of the corpus. Optionally include also names of others involved in the transcription or elaboration of the text, sponsors, and funding agencies. The name of the person responsible for physical data input need not normally be recorded, unless that person is also intellectually responsible for some aspect of the creation of the file.

Where the person whose responsibility is to be documented is not an author, sponsor, funding body, or principal researcher, the <respStmt> element should be used. This has two subcomponents: a <name> element identifying a responsible individual or organization, and a <resp> element indicating the nature of the responsibility. No specific recommendations are made at this time as to appropriate content for the <resp>: it should make clear the nature of the responsibility concerned, as in the examples below.

Names given may be personal names or corporate names. Give all names in the form in which the persons or bodies wish to be publicly cited. This would usually be the fullest form of the name, including first names.[2]

Examples:

```
<titleStmt>
  <title>Capgrave's Life of St. John Norbert: a
     machine-readable transcription</title>
  <respStmt>
    <resp>compiled by</resp>
    <name>P.J. Lucas</name>
  </respStmt>
</titleStmt>
```

```
<titleStmt>
  <title>Two stories by Edgar Allen Poe: electronic version</title>
  <author>Poe, Edgar Allen (1809-1849)</author>
  <respStmt>
    <resp>compiled by</resp>
    <name>James D. Benson</name>
  </respStmt>
</titleStmt>
```

[2] Agencies compiling catalogues of machine-readable files are recommended to use available authority lists, such as the Library of Congress Name Authority List, for all common personal names.

```
<titleStmt>
  <title>Yogadarśanam (arthāt
     yogasūtrapūṭhaḥ):
     a digital edition.</title>
  <title>The Yogasūtras of Patañjali:
     a digital edition.</title>
  <funder>Wellcome Institute for the History of Medicine</funder>
  <principal>Dominik Wujastyk</principal>
  <respStmt>
    <name>Wieslaw Mical</name>
    <resp>data entry and proof correction</resp>
  </respStmt>
  <respStmt>
    <name>Jan Hajic</name>
    <resp>conversion to TEI-conformant markup</resp>
  </respStmt>
</titleStmt>
```

2.2.2 The Edition Statement

The <editionStmt> element is the second component of the <fileDesc> element. It is optional but recommended.

<editionStmt> (edition statement) groups information relating to one edition of a text.

It contains either phrases or more specialized elements identifying the edition and those responsible for it:

<edition> (edition) describes the particularities of one edition of a text.

<respStmt> (statement of responsibility) supplies a statement of responsibility for the intellectual content of a text, edition, recording, or series, where the specialized elements for authors, editors, etc. do not suffice or do not apply.

<name> (name, proper noun) contains a proper noun or noun phrase.

<resp> (responsibility) contains a phrase describing the nature of a person's intellectual responsibility.

For printed texts, the word *edition* applies to the set of all the identical copies of an item produced from one master copy and issued by a particular publishing agency or a group of such agencies. A change in the identity of the distributing body or bodies does not normally constitute a change of edition, while a change in the master copy does.

For electronic texts, the notion of a 'master copy' is not entirely appropriate, since they are far more easily copied and modified than printed ones; nonetheless the term *edition* may be used for a particular state of a machine-readable text at which substantive changes are made and fixed. Synonymous terms used in these Guidelines are *version*, *level*, and *release*. The words *revision* and *update*, by contrast, are used for minor changes to a file which do not amount to a new edition.

No simple rule can specify how 'substantive' changes have to be before they are regarded as producing a new edition, rather than a simple update. The general principle proposed here is that the production of a new edition entails a significant change in the intellectual content of the file, rather than its encoding or appearance. The addition of analytic coding to a text would thus constitute a new edition, while automatic conversion from one coded representation to another would not. Changes relating to the character code or physical storage details, corrections of misspellings, simple changes in the arrangement of the contents and changes in the output format do not normally constitute a new edition, whereas the addition of new information (e.g. a linguistic analysis expressed in part-of-speech tagging, sound or graphics, referential links to external data sets) almost always does.

Clearly, there will always be borderline cases and the matter is somewhat arbitrary. The simplest rule is: if you think that your file is a new edition, then call it such. An edition statement is optional for the first release of a computer file; it is mandatory for each later release, though this requirement cannot be enforced by the parser.

Note that *all* changes in a file, whether or not they are regarded as constituting a new edition or simply a new revision, should be independently noted in the revision description section of the file header (see section *2.5. The Revision Description*).

The <edition> element should contain phrases describing the edition or version, including the word *edition*, *version*, or equivalent, together with a number or date, or terms indicating difference from other editions such as *new edition*, *revised edition* etc. Any dates that occur within the edition statement should be marked with the <date> element. The **n** attribute of the <edition> element may be used as elsewhere to supply any formal identification (such as a version number) for the edition.

One or more <respStmt> elements may also be used to supply statements of responsibility for the edition in question. These may refer to individuals or corporate bodies and can indicate functions such as that of a reviser, or can name the person or body responsible for the provision of supplementary matter, of appendices, etc., in a new edition. For further detail on the <respStmt> element, see section *3.11. Bibliographic Citations and References*.

Some examples follow:

```
<editionStmt>
  <edition  n="P2">Second draft, substantially
     extended, revised, and corrected.</edition>
</editionStmt>
```

```
<editionStmt>
  <edition>Student's edition, <date>June 1987</date>
  </edition>
  <respStmt>
    <resp>New annotations by</resp>
    <name>George Brown</name>
  </respStmt>
</editionStmt>
```

2.2.3 Type and Extent of File

The <extent> element is the third component of the <fileDesc> element. It is optional.

<extent> describes the approximate size of a text as stored on some carrier medium, whether digital or non-digital, specified in any convenient units.

For printed books, information about the carrier, such as the kind of medium used and its size, are of great importance in cataloguing procedures. The print-oriented rules for bibliographic description of an item's medium and extent need some re-interpretation when applied to electronic media. An electronic file exists as a distinct entity quite independently of its carrier and remains the same intellectual object whether it is stored on a magnetic tape, a CD-ROM, a set of floppy disks, or as a file on a mainframe computer. Since, moreover, these Guidelines are specifically aimed at facilitating transparent document storage and interchange, any purely machine-dependent information should be irrelevant as far as the file header is concerned.

This is particularly true of information about file-type although library-oriented rules for cataloguing often distinguish two types of computer file: 'data' and 'programs'. This distinction is quite difficult to draw in some cases, for example, hypermedia or texts with built in search and retrieval software.

Although it is equally system-dependent, some measure of the size of the computer file may be of use for cataloguing and other practical purposes. Because the measurement and expression of file size is fraught with difficulties, only very general recommendations are possible; the element <extent> is provided for this purpose. It contains a phrase indicating the size or approximate size of the computer file in one of the following ways:

- in bytes of a specified length (e.g. '4000 16-bit bytes')
- as falling within a range of categories, for example:
  - less than 1 Mb
  - between 1 Mb and 5 Mb
  - between 6 Mb and 10 Mb
  - over 10 Mb
- in terms of any convenient logical units (for example, words or sentences, citations, paragraphs)
- in terms of any convenient physical units (for example, blocks, disks, tapes)

The use of standard abbreviations for units of quantity is recommended where applicable, here as elsewhere (see `http://physics.nist.gov/cuu/Units/binary.html`).

Examples:

```
<extent>between 1 16-bit MB and 2 16-bit MB</extent>
<extent>4.2 MiB</extent>
<extent>4532 bytes</extent>
<extent>3200 sentences</extent>
<extent>5 90 mm High Density Diskettes</extent>
```

2.2.4 Publication, Distribution, etc.

The <publicationStmt> element is the fourth component of the <fileDesc> element and is mandatory.

<publicationStmt> (publication statement) groups information concerning the publication or distribution of an electronic or other text.

It may contain either a simple prose description organized as one or more paragraphs, or one or more elements from the model.publicationStmt class. This class groups a number of elements which are discussed in order below.

<publisher> provides the name of the organization responsible for the publication or distribution of a bibliographic item.

<distributor> supplies the name of a person or other agency responsible for the distribution of a text.

<authority> (release authority) supplies the name of a person or other agency responsible for making an electronic file available, other than a publisher or distributor.

The publisher is the person or institution by whose authority a given edition of the file is made public. The distributor is the person or institution from whom copies of the text may be obtained. Where a text is not considered formally published, but is nevertheless made available for circulation by some individual or organization, this person or institution is termed the release authority.

At least one of the above three elements must be present, unless the entire publication statement is given as prose. Each may be followed by one or more of the following elements, in the following order:[3]

<pubPlace> (publication place) contains the name of the place where a bibliographic item was published.

[3] This constraint is not however enforced by the current version of the TEI Guidelines.

<address> contains a postal address, for example of a publisher, an organization, or an individual.

<idno> (identifying number) supplies any standard or non-standard number used to identify a bibliographic item.

@**type** categorizes the number, for example as an ISBN or other standard series.

<availability> supplies information about the availability of a text, for example any restrictions on its use or distribution, its copyright status, etc.

@**status** supplies a code identifying the current availability of the text.

<date> contains a date in any format.

Note that the dates, places, etc., given in the publication statement relate to the publisher, distributor, or release authority most recently mentioned. If the text was created at some date other than its date of publication, its date of creation should be given within the <profileDesc> element, not in the publication statement. Give any other useful dates (e.g., dates of collection of data) in a note.

Additional detailed elements may be used for the encoding of names, dates, and addresses, as further described in section *3.5. Names, Numbers, Dates, Abbreviations, and Addresses* when the module described in chapter *13. Names, Dates, People, and Places* is included in a schema.

Examples:

```
<publicationStmt>
  <publisher>Oxford University Press</publisher>
  <pubPlace>Oxford</pubPlace>
  <date>1989</date>
  <idno  type="ISBN">0-19-254705-4</idno>
  <availability>
    <p>Copyright 1989, Oxford University Press</p>
  </availability>
</publicationStmt>
```

```
<publicationStmt>
  <authority>James D. Benson</authority>
  <pubPlace>London</pubPlace>
  <date>1984</date>
</publicationStmt>
```

```
<publicationStmt>
  <publisher>Sigma Press</publisher>
  <address>
    <addrLine>21 High Street,</addrLine>
    <addrLine>Wilmslow,</addrLine>
    <addrLine>Cheshire M24 3DF</addrLine>
  </address>
  <date>1991</date>
  <distributor>Oxford Text Archive</distributor>
  <idno  type="ota">1256</idno>
  <availability>
    <p>Available with prior consent of depositor for
        purposes of academic research and teaching only.</p>
  </availability>
</publicationStmt>
```

2.2.5 The Series Statement

The <seriesStmt> element is the fifth component of the <fileDesc> element and is optional.

<seriesStmt> (series statement) groups information about the series, if any, to which a publication belongs.

In bibliographic parlance, a series may be defined in one of the following ways:

- A group of separate items related to one another by the fact that each item bears, in addition to its own title proper, a collective title applying to the group as a whole. The individual items may or may not be numbered.
- Each of two or more volumes of essays, lectures, articles, or other items, similar in character and issued in sequence.
- A separately numbered sequence of volumes within a series or serial.

The <seriesStmt> element may contain a prose description or one or more of the following more specific elements:

<title> contains the full title of a work of any kind.

<idno> (identifying number) supplies any standard or non-standard number used to identify a bibliographic item.

<respStmt> (statement of responsibility) supplies a statement of responsibility for the intellectual content of a text, edition, recording, or series, where the specialized elements for authors, editors, etc. do not suffice or do not apply.

<resp> (responsibility) contains a phrase describing the nature of a person's intellectual responsibility.

<name> (name, proper noun) contains a proper noun or noun phrase.

The <idno> may be used to supply any identifying number associated with the item, including both standard numbers such as an ISSN and particular issue numbers. (Arabic numerals separated by punctuation are recommended for this purpose: **6.19.33**, for example, rather than **VI/xix:33**). Its **type** attribute is used to categorize the number further, taking the value **ISSN** for an ISSN for example.

Examples:

```
<seriesStmt>
  <title  level="s">Machine-Readable Texts for the Study of
     Indian Literature</title>
  <respStmt>
    <resp>ed.  by</resp>
    <name>Jan Gonda</name>
  </respStmt>
  <idno  type="vol">1.2</idno>
  <idno  type="ISSN">0 345 6789</idno>
</seriesStmt>
```

2.2.6 The Notes Statement

The <notesStmt> element is the sixth component of the <fileDesc> element and is optional. If used, it contains one or more <note> elements, each containing a single piece of descriptive information of the kind treated as 'general notes' in traditional bibliographic descriptions.

<notesStmt> (notes statement) collects together any notes providing information about a text additional to that recorded in other parts of the bibliographic description.

<note> contains a note or annotation.

Some information found in the notes area in conventional bibliography has been assigned specific elements in these Guidelines; in particular the following items should be tagged as indicated, rather than as general notes:

- the nature, scope, artistic form, or purpose of the file; also the genre or other intellectual category to which it may belong: e.g. 'Text types: newspaper editorials and reportage, science fiction, westerns, and detective stories'. These should be formally described within the <profileDesc> element (section *2.4. The Profile Description*).
- summary description providing a factual, non-evaluative account of the subject content of the file: e.g. 'Transcribes interviews on general topics with native speakers of English in 17 cities during the spring and summer of 1963.' These should also be formally described within the <profileDesc> element (section *2.4. The Profile Description*).
- bibliographic details relating to the source or sources of an electronic text: e.g. 'Transcribed from the Norton facsimile of the 1623 Folio'. These should be formally described in the <sourceDesc> element (section *2.2.7. The Source Description*).
- further information relating to publication, distribution, or release of the text, including sources from which the text may be obtained, any restrictions on its use or formal terms on its availability. These should be placed in the appropriate division of the <publicationStmt> element (section *2.2.4. Publication, Distribution, etc.*).
- publicly documented numbers associated with the file: e.g. 'ICPSR study number 1803' or 'Oxford Text Archive text number 1243'. These should be placed in an <idno> element within the appropriate division of the <publicationStmt> element. International Standard Serial Numbers (ISSN), International Standard Book Numbers (ISBN), and other internationally agreed upon standard numbers that uniquely identify an item, should be treated in the same way, rather than as specialized bibliographic notes.

Nevertheless, the <notesStmt> element may be used to record potentially significant details about the file and its features, e.g.:

- dates, when they are relevant to the content or condition of the computer file: e.g. 'manual dated 1983', 'Interview wave I: Apr. 1989; wave II: Jan. 1990'
- names of persons or bodies connected with the technical production, administration, or consulting functions of the effort which produced the file, if these are not named in statements of responsibility in the title or edition statements of the file description: e.g. 'Historical commentary provided by Mark Cohen'
- availability of the file in an additional medium or information not already recorded about the availability of documentation: e.g. 'User manual is loose-leaf in eleven paginated sections'
- language of work and abstract, if not encoded in the <langUsage> element, e.g. 'Text in English with summaries in French and German'
- The unique name assigned to a serial by the International Serials Data System (ISDS), if not encoded in an <idno>
- lists of related publications, either describing the source itself, or concerned with the creation or use of the electronic work, e.g. 'Texts used in Burrows (1987)'

Each such item of information may be tagged using the general-purpose <note> element, which is described in section *3.8. Notes, Annotation, and Indexing.* Groups of notes are contained within the <notesStmt> element, as in the following example:

```
<notesStmt>
  <note>Historical commentary provided by Mark Cohen.</note>
```

```
  <note>OCR scanning done at University of Toronto.</note>
</notesStmt>
```

There are advantages, however, to encoding such information with more precise elements elsewhere in the TEI header, when such elements are available. For example, the notes above might be encoded as follows:

```
<titleStmt>
  <title>…</title>
  <respStmt>
    <persName>Mark Cohen</persName>
    <resp>historical commentary</resp>
  </respStmt>
  <respStmt>
    <orgName>University of Toronto</orgName>
    <resp>OCR scanning</resp>
  </respStmt>
</titleStmt>
```

2.2.7 The Source Description

The <sourceDesc> element is the seventh and final component of the <fileDesc> element. It is a mandatory element and is used to record details of the source or sources from which a computer file is derived. This might be a printed text or manuscript, another computer file, an audio or video recording of some kind, or a combination of these. An electronic file may also have no source, if what is being catalogued is an original text created in electronic form.

<sourceDesc> (source description) supplies a description of the source text(s) from which an electronic text was derived or generated.

The <sourceDesc> element may contain little more than a simple prose description, or a brief note stating that the document has no source:

```
<sourceDesc>
  <p>Born digital.</p>
</sourceDesc>
```

Alternatively, it may contain elements drawn from the following three classes:

model.biblLike groups elements containing a bibliographic description.

model.sourceDescPart groups elements which may be used inside <sourceDesc> and appear multiple times.

model.listLike groups list-like elements.

These classes make available by default a range of ways of providing bibliographic citations which specify the provenance of the text. For written or printed sources, the source may be described in the same way as any other bibliographic citation, using one of the following elements:

<bibl> (bibliographic citation) contains a loosely-structured bibliographic citation of which the sub-components may or may not be explicitly tagged.

<biblStruct> (structured bibliographic citation) contains a structured bibliographic citation, in which only bibliographic sub-elements appear and in a specified order.

<listBibl> (citation list) contains a list of bibliographic citations of any kind.

These elements are described in more detail in section *3.11. Bibliographic Citations and References.* Using them, a source might be described in very simple terms:

```
<sourceDesc>
  <bibl>The first folio of Shakespeare, prepared by
     Charlton Hinman (The Norton Facsimile, 1968)</bibl>
</sourceDesc>
```

or with more elaboration:

```
<sourceDesc>
  <biblStruct  xml:lang="fr">
    <monogr>
      <author>Eugène Sue</author>
      <title>Martin, l'enfant trouvé</title>
      <title  type="sub">Mémoires d'un valet de chambre</title>
      <imprint>
        <pubPlace>Bruxelles et Leipzig</pubPlace>
        <publisher>C. Muquardt</publisher>
        <date  when="1846">1846</date>
      </imprint>
    </monogr>
  </biblStruct>
</sourceDesc>
```

When the header describes a text derived from some pre-existing TEI-conformant or other digital document, it may be simpler to use the following element:

<biblFull> (fully-structured bibliographic citation) contains a fully-structured bibliographic citation, in which all components of the TEI file description are present.

since this is designed specifically for documents derived from texts which were 'born digital', as further discussed in section *2.2.8. Computer Files Derived from Other Computer Files* .

When the module for manuscript description is included in a schema, this class also makes available the following element:

<msDesc> (manuscript description) contains a description of a single identifiable manuscript.

which enables the encoder to record very detailed information about one or more manuscript or analogous sources, as further discussed in *10. Manuscript Description*.

The model.sourceDescPart class also makes available additional elements when additional modules are included. For example, when the spoken module is included, the <sourceDesc> element may also include the following special-purpose elements, intended for cases where an electronic text is derived from a spoken text rather than a written one:

<scriptStmt> (script statement) contains a citation giving details of the script used for a spoken text.

<recordingStmt> (recording statement) describes a set of recordings used as the basis for transcription of a spoken text.

Full descriptions of these elements and their contents are given in section *8.2. Documenting the Source of Transcribed Speech*.

The source description may also include lists of names, persons, places, etc. when these are considered to form part of the source for an encoded document. When such information is recorded using the specialized elements discussed in the namesdates module (*13. Names, Dates, People, and Places*), the class model.listLike makes available the following elements to hold such information:

<listNym> (list of canonical names) contains a list of nyms, that is, standardized names for any thing.

<listOrg> (list of organizations) contains a list of elements, each of which provides information about an identifiable organization.

<listPerson> (list of persons) contains a list of descriptions, each of which provides information about an identifiable person or a group of people, for example the participants in a language interaction, or the people referred to in a historical source.

<listPlace> (list of places) contains a list of places, optionally followed by a list of relationships (other than containment) defined amongst them.

2.2.8 Computer Files Derived from Other Computer Files

If a computer file (call it B) is derived not from a printed source but from another computer file (call it A) which includes a TEI file header, then the source text of computer file B is another computer file, A. The four sections of A's file header will need to be incorporated into the new header for B in slightly differing ways, as listed below:

fileDesc A's file description should be copied into the <sourceDesc> section of B's file description, enclosed within a <biblFull> element

profileDesc A's <profileDesc> should be copied into B's, in principle unchanged; it may however be expanded by project-specific information relating to B.

encodingDesc A's encoding practice may or (more likely) may not be the same as B's. Since the object of the encoding description is to define the relationship between the current file and its source, in principle only changes in encoding practice between A and B need be documented in B. The relationship between A and its source(s) is then only recoverable from the original header of A. In practice it may be more convenient to create a new complete <encodingDesc> for B based on A's.

revisionDesc B is a new computer file, and should therefore have a new revision description. If, however, it is felt useful to include some information from A's <revisionDesc>, for example dates of major updates or versions, such information must be clearly marked as relating to A rather than to B.

This concludes the discussion of the <fileDesc> element and its contents.

2.3 The Encoding Description

The <encodingDesc> element is the second major subdivision of the TEI header. It specifies the methods and editorial principles which governed the transcription or encoding of the text in hand and may also include sets of coded definitions used by other components of the header. Though not formally required, its use is highly recommended.

<encodingDesc> (encoding description) documents the relationship between an electronic text and the source or sources from which it was derived.

The encoding description may contain paragraphs of text, marked up using the <p> element, or it may contain more specialised elements taken from the model.encodingPart class. By default, this class makes available the following elements:

<projectDesc> (project description) describes in detail the aim or purpose for which an electronic file was encoded, together with any other relevant information concerning the process by which it was assembled or collected.

<samplingDecl> (sampling declaration) contains a prose description of the rationale and methods used in sampling texts in the creation of a corpus or collection.

<editorialDecl> (editorial practice declaration) provides details of editorial principles and practices applied during the encoding of a text.

<tagsDecl> (tagging declaration) provides detailed information about the tagging applied to a document.

<refsDecl> (references declaration) specifies how canonical references are constructed for this text.

<classDecl> (classification declarations) contains one or more taxonomies defining any classificatory codes used elsewhere in the text.

<appInfo> (application information) records information about an application which has edited the TEI file.

Each of these elements is further described in the appropriate section below. Other modules have the ability to extend this class; examples are noted in section *2.3.8. Module-Specific Declarations*

2.3.1 The Project Description

The <projectDesc> element may be used to describe, in prose, the purpose for which a digital resource was created, together with any other relevant information concerning the process by which it was assembled or collected. This is of particular importance for corpora or miscellaneous collections, but may be of use for any text, for example to explain why one kind of encoding practice has been followed rather than another.

<projectDesc> (project description) describes in detail the aim or purpose for which an electronic file was encoded, together with any other relevant information concerning the process by which it was assembled or collected.

For example:

```
<encodingDesc>
  <projectDesc>
    <p>Texts collected for use in the
        Claremont Shakespeare Clinic, June 1990.</p>
  </projectDesc>
</encodingDesc>
```

2.3.2 The Sampling Declaration

The <samplingDecl> element may be used to describe, in prose, the rationale and methods used in selecting texts, or parts of text, for inclusion in the resource.

<samplingDecl> (sampling declaration) contains a prose description of the rationale and methods used in sampling texts in the creation of a corpus or collection.

It should include information about such matters as

- the size of individual samples
- the method or methods by which they were selected
- the underlying population being sampled
- the object of the sampling procedure used

but is not restricted to these.

```
<samplingDecl>
  <p>Samples of 2000 words taken from the beginning of the text.</p>
</samplingDecl>
```

It may also include a simple description of any parts of the source text included or excluded.

```
<samplingDecl>
  <p>Text of stories only has been transcribed. Pull quotes, captions,
     and advertisements have been silently omitted. Any mathematical
     expressions requiring symbols not present in the ISOnum or ISOpub
     entity sets have been omitted, and their place marked with a GAP
     element.</p>
</samplingDecl>
```

A sampling declaration which applies to more than one text or division of a text need not be repeated in the header of each such text. Instead, the **decls** attribute of each text (or subdivision of the text) to which the sampling declaration applies may be used to supply a cross-reference to it, as further described in section *15.3. Associating Contextual Information with a Text*.

2.3.3 The Editorial Practices Declaration

The <editorialDecl> element is used to provide details of the editorial practices applied during the encoding of a text.

<editorialDecl> (editorial practice declaration) provides details of editorial principles and practices applied during the encoding of a text.

It may contain a prose description only, or one or more of a set of specialized elements, members of the TEI model.editorialDeclPart class. Where an encoder wishes to record an editorial policy not specified above, this may be done by adding a new element to this class, using the mechanisms discussed in chapter *23.2. Personalization and Customization*.

Some of these policy elements carry attributes to support automated processing of certain well-defined editorial decisions; all of them contain a prose description of the editorial principles adopted with respect to the particular feature concerned. Examples of the kinds of questions which these descriptions are intended to answer are given in the list below.

<correction>

<correction> (correction principles) states how and under what circumstances corrections have been made in the text.
@**status** indicates the degree of correction applied to the text.
@**method** indicates the method adopted to indicate corrections within the text.

Was the text corrected during or after data capture? If so, were corrections made silently or are they marked using the tags described in section *3.4. Simple Editorial Changes*? What principles have been adopted with respect to omissions, truncations, dubious corrections, alternate readings, false starts, repetitions, etc.?

<normalization>

<normalization> indicates the extent of normalization or regularization of the original source carried out in converting it to electronic form.
@**source** indicates the authority for any normalization carried out.
@**method** indicates the method adopted to indicate normalizations within the text.

Was the text normalized, for example by regularizing any non-standard spellings, dialect forms, etc.? If so, were normalizations performed silently or are they marked using the tags described in section *3.4. Simple Editorial Changes*? What authority was used for the regularization? Also, what principles

were used when normalizing numbers to provide the standard values for the **value** attribute described in section *3.5.3. Numbers and Measures* and what format used for them?

<quotation>

<quotation> specifies editorial practice adopted with respect to quotation marks in the original.
@**marks** (quotation marks) indicates whether or not quotation marks have been retained as content within the text.
@**form** specifies how quotation marks are indicated within the text.

How were quotation marks processed? Are apostrophes and quotation marks distinguished? How? Are quotation marks retained as content in the text or replaced by markup? Are there any special conventions regarding for example the use of single or double quotation marks when nested? Is the file consistent in its practice or has this not been checked?

<hyphenation>

<hyphenation> summarizes the way in which hyphenation in a source text has been treated in an encoded version of it.
@**eol** (end-of-line) indicates whether or not end-of-line hyphenation has been retained in a text.

Does the encoding distinguish 'soft' and 'hard' hyphens? What principle has been adopted with respect to end-of-line hyphenation where source lineation has not been retained? Have soft hyphens been silently removed, and if so what is the effect on lineation and pagination?

<segmentation>

<segmentation> describes the principles according to which the text has been segmented, for example into sentences, tone-units, graphemic strata, etc.

How is the text segmented? If <s> or <seg> segmentation units have been used to divide up the text for analysis, how are they marked and how was the segmentation arrived at?

<stdVals>

<stdVals> (standard values) specifies the format used when standardized date or number values are supplied.

In most cases, attributes bearing standardized values (such as the **when** or **when-iso** attribute on dates) should conform to a defined W3C or ISO datatype. In cases where this is not appropriate, this element may be used to describe the standardization methods underlying the values supplied.

<interpretation>

<interpretation> describes the scope of any analytic or interpretive information added to the text in addition to the transcription.

Has any analytic or 'interpretive' information been provided — that is, information which is felt to be non-obvious, or potentially contentious? If so, how was it generated? How was it encoded? If feature-structure analysis has been used, are <fsdDecl> elements (section *18.11. Feature System Declaration*) present?

Any information about the editorial principles applied not falling under one of the above headings should be recorded in a distinct list of items. Experience shows that a full record should be kept of decisions relating to editorial principles and encoding practice, both for future users of the text and for the project which produced the text in the first instance. Some simple examples follow:

```
<editorialDecl>
  <segmentation>
    <p>
      <gi>s</gi> elements mark orthographic sentences and
        are numbered sequentially
        within their parent <gi>div</gi> element
    </p>
  </segmentation>
  <interpretation>
    <p>The part of speech analysis applied throughout section 4 was
        added by hand and has not been checked.</p>
  </interpretation>
  <correction>
    <p>Errors in transcription controlled by using the
        WordPerfect spelling checker.</p>
  </correction>
  <normalization  source="http://szotar.sztaki.hu/webster/">
    <p>All words converted to Modern American spelling following
        Websters 9th Collegiate dictionary.</p>
  </normalization>
  <quotation  marks="all"  form="std">
    <p>All opening quotation marks represented by entity reference
    <ident  type="ge">odq</ident>; all closing quotation marks
        represented by entity reference <ident  type="ge">cdq</ident>.</p>
  </quotation>
</editorialDecl>
```

An editorial practices declaration which applies to more than one text or division of a text need not be repeated in the header of each such text. Instead, the **decls** attribute of each text (or subdivision of the text) to which it applies may be used to supply a cross-reference to it, as further described in section *15.3. Associating Contextual Information with a Text*.

2.3.4 The Tagging Declaration

The <tagsDecl> element is used to record the following information about the tagging used within a particular text:

- the namespace to which elements appearing within the transcribed text belong.
- how often particular elements appear within the text, so that a recipient can validate the integrity of a text during interchange.
- any comment relating to the usage of particular elements not specified elsewhere in the header.
- a default rendition applicable to all instances of an element.

This information is conveyed by the following elements:

<rendition> supplies information about the rendition or appearance of one or more elements in the source text.

@**scheme** identifies the language used to describe the rendition.

<namespace> supplies the formal name of the namespace to which the elements documented by its children belong.

<tagUsage> supplies information about the usage of a specific element within a text.

The <tagsDecl> element consists of an optional sequence of <rendition> elements, each of which must bear a unique identifier, followed by an optional sequence of one or more <namespace> elements, containing a series

of <tagUsage> elements, one for each distinct element from that namespace occurring within the outermost <text> element of a TEI document.

2.3.4.1 Rendition

The <rendition> element allows the encoder to specify how one or more elements are rendered in the original source in any of the following ways:

- using an informal prose description
- using a standard stylesheet language such as CSS or XSL-FO
- using a project-defined formal language

One or more such specifications may be associated with elements of a document in two ways:

- the **render** attribute of the appropriate <tagUsage> element may be used to indicate a default rendition for all occurrences of the named element
- the global **rendition** attribute may be used on any element to indicate its rendition, over-riding any supplied default value

The global **rend** attribute may also be used to supply an informal description of the rendering for an element; if this is supplied in addition to the **rendition** attribute it takes precedence, just as it also overrides any default specified for that element.

For example, the following schematic shows how an encoder might specify that all <p> elements are by default to be rendered using one set of specifications identified as **style1**, while <hi> elements are to use a different set, identified as **style2**:

```
<tagsDecl>
  <rendition  xml:id="style1">
     ... description of one default rendition here ...
  </rendition>
  <rendition  xml:id="style2">
     ... description of another default rendition here ...
  </rendition>
  <namespace  name="http://www.tei-c.org/ns/1.0">
    <tagUsage  gi="p"  render="#style1"> ... </tagUsage>
    <tagUsage  gi="hi"  render="#style2"> ... </tagUsage>
  </namespace>
</tagsDecl>
<!-- elsewhere in the document -->
<p>This paragraph,mostly rendered in style1, contains a few words
<hi>rendered in style2</hi>
</p>
<p  rendition="#style2">This paragraph is all rendered in style2</p>
<p>This is back to style1</p>
```

As noted above, the content of the <rendition> element may describe the appearance of the source material using prose, a project-defined formal language, or either of the existing standard languages: the Cascading Stylesheet Language (Lie and Bos (eds.) (1999)) and the XML vocabulary for specifying formatting semantics which forms a part of the W3C's Extensible Stylesheet Language (Berglund (ed.) (2006)). The **scheme** attribute indicates which of these applies to a given <rendition> element, and takes the following values:

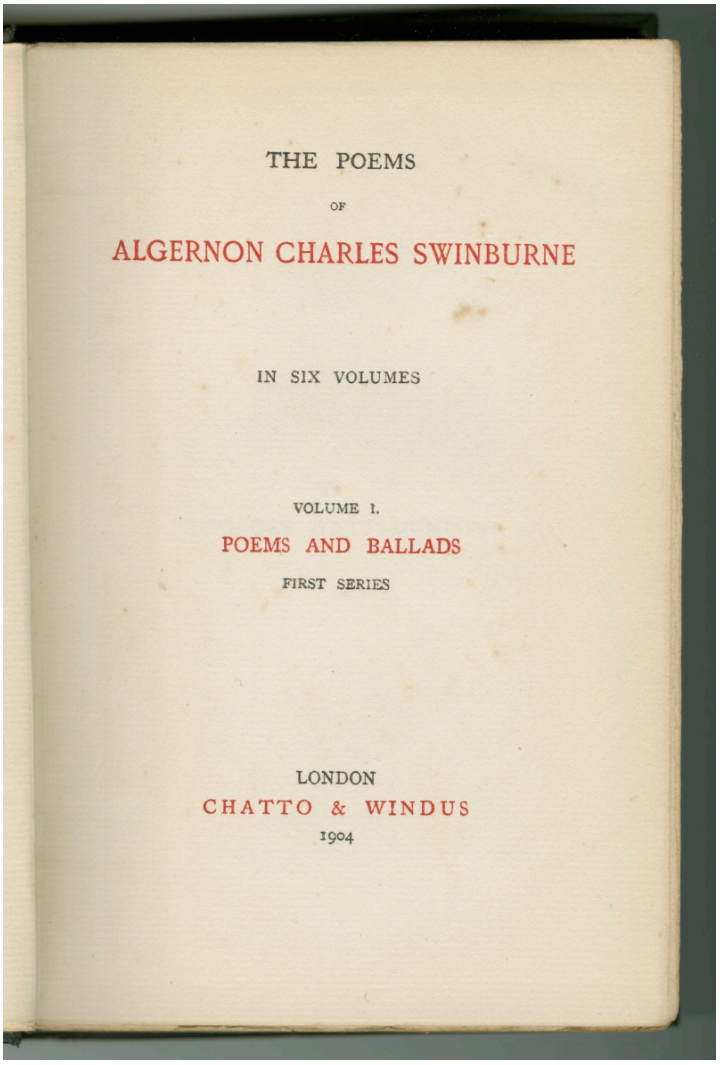
free Informal free text description

css Cascading Stylesheet Language

xslfo Extensible Stylesheet Language Formatting Objects

other A user-defined formal description language

In the following extended example we consider how best to capture the appearance of a typical early 20th century titlepage, such as that in the following figure: Elements for the encoding of the information on a



titlepage are presented in *4.6. Title Pages*; here we consider how we might go about encoding some of the visual information as well, using the <rendition> element and its corresponding attribute.

First we define a rendition element for each aspect of the source page rendition that we wish to retain. Details of CSS are given in Lie and Bos (eds.) (1999); we use it here simply to provide a vocabulary with which to describe such aspects as font size and style, letter and line spacing, and colour.

```
<tagsDecl>
  <rendition xml:id="center" scheme="css">text-align: center;</rendition>
  <rendition xml:id="small" scheme="css">font-size:
     small;</rendition>
  <rendition xml:id="large" scheme="css">font-size: large;</rendition>
  <rendition xml:id="x-large" scheme="css">font-size: x-large;</rendition>
  <rendition xml:id="xx-large" scheme="css">font-size: xx-large</rendition>
  <rendition xml:id="expanded" scheme="css">letter-spacing: +3pt;</rendition>
```

```
  <rendition  xml:id="x-space"  scheme="css">line-height: 150%;</rendition>
  <rendition  xml:id="xx-space"  scheme="css">line-height: 200%;</rendition>
  <rendition  xml:id="red"  scheme="css">color: red;</rendition>
</tagsDecl>
```

The global **rendition** attribute can now be used to specify on any element which of the above rendition features apply to it. For example, a title page might be encoded as follows:

```
<titlePage>
  <docTitle  rendition="#center #x-space">
    <titlePart>
      <lb/>
      <hi  rendition="#x-large">THE POEMS</hi>
      <lb/>
      <hi  rendition="#small">OF</hi>
      <lb/>
      <hi  rendition="#red #xx-large">ALGERNON CHARLES SWINBURNE</hi>
      <lb/>
      <hi  rendition="#large #xx-space">IN SIX VOLUMES</hi>
    </titlePart>
    <titlePart  rendition="#xx-space"> VOLUME I.
    <lb/>
      <hi  rendition="#red #x-large">POEMS AND BALLADS</hi>
      <lb/>
      <hi  rendition="#x-space">FIRST SERIES</hi>
    </titlePart>
  </docTitle>
  <docImprint  rendition="#center">
    <lb/>
    <pubPlace  rendition="#xx-space">LONDON</pubPlace>
    <lb/>
    <publisher  rendition="#red #expanded">CHATTO & WINDUS</publisher>
    <lb/>
    <docDate  when="1904"  rendition="#small">1904</docDate>
  </docImprint>
</titlePage>
```

Source: [189]

2.3.4.2 Tag usage

As noted above, each <namespace> element, if present, should contain exactly one occurrence of a <tagUsage> element for each distinct element from the given namespace that occurs within the outermost <text> element associated with the <teiHeader> in which it appears.[4] The <tagUsage> element is used to supply a count of the number of occurrences of this element within the text, which is given as the value of its **occurs** attribute. It may also be used to hold any additional usage information, which is supplied as running prose within the element itself.

For example:

```
<tagUsage  gi="hi"  occurs="28"> Used only to mark English words italicised in the copy text.
</tagUsage>
```

[4] In the case of a TEI corpus (*15. Language Corpora*), a <tagsDecl> in a corpus header will describe tag usage across the whole corpus, while one in an individual text header will describe tag usage for the individual text concerned.

This indicates that the <hi> element appears a total of 28 times in the <text> element in question, and that the encoder has used it to mark italicised English words only.

The **withId** attribute may optionally be used to specify how many of the occurrences of the element in question bear a value for the global **xml:id** attribute, as in the following example:

```
<tagUsage gi="pb" occurs="321" withId="321"> Marks page breaks in the York (1734) edition only
</tagUsage>
```

This indicates that the <pb> element occurs 321 times, on each of which an identifier is provided.

The content of the <tagUsage> element is not susceptible of automatic processing. It should not therefore be used to hold information for which provision is already made by other components of the encoding description. A TEI conformant document is not required to provide any <tagUsage> elements, but if it does, then TEI recommended practice is to provide <namespace> and <tagUsage> elements for each distinct element and namespace used in the associated text. If, in addition, counts are specified by the **occurs** attributes, these must correspond with the number of such elements present in the document.

2.3.5 The Reference System Declaration

The <refsDecl> element is used to document the way in which any standard referencing scheme built into the encoding works. It may contain either a series of prose paragraphs or the following specialized elements:

<refsDecl> (references declaration) specifies how canonical references are constructed for this text.

<cRefPattern> (canonical reference pattern) specifies an expression and replacement pattern for transforming a canonical reference into a URI.

<refState/> (reference state) specifies one component of a canonical reference defined by the milestone method.

Note that not all possible referencing schemes are equally easily supported by current software systems. A choice must be made between the convenience of the encoder and the likely efficiency of the particular software applications envisaged, in this context as in many others. For a more detailed discussion of referencing systems supported by these Guidelines, see section *3.10. Reference Systems* below.

A referencing scheme may be described in one of three ways using this element:

- as a prose description
- as a series of pairs of regular expressions and XPaths
- as a concatenation of sequentially organized milestones

Each method is described in more detail below. Only one method can be used within a single <refsDecl> element.

More than one <refsDecl> element can be included in the header if more than one canonical reference scheme is to be used in the same document, but the current proposals do not check for mutual inconsistency.

2.3.5.1 Prose Method

The referencing scheme may be specified within the <refsDecl> by a simple prose description. Such a description should indicate which elements carry identifying information, and whether this information is represented as attribute values or as content. Any special rules about how the information is to be interpreted when reading or generating a reference string should also be specified here. Such a prose description cannot be processed automatically, and this method of specifying the structure of a canonical reference system is therefore not recommended for automatic processing.

For example:

```
<refsDecl>
  <p>The <att>n</att> attribute of each text in this corpus carries a
     unique identifying code for the whole text. The title of the text is
     held as the content of the first <gi>head</gi> element within each
     text. The <att>n</att> attribute on each <gi>div1</gi> and
  <gi>div2</gi> contains the canonical reference for each such
     division, in the form 'XX.yyy', where XX is the book number in Roman
     numerals, and yyy the section number in arabic. Line breaks are
     marked by empty <gi>lb</gi> elements, each of which includes the
     through line number in Casaubon's edition as the value of its
  <gi>n</gi> attribute.</p>
  <p>The through line number and the text identifier uniquely identify
     any line. A canonical reference may be made up by concatenating the
  <gi>n</gi> values from the <gi>text</gi>, <gi>div1</gi>, or
  <gi>div2</gi> and calculating the line number within each part.</p>
</refsDecl>
```

2.3.5.2 Search-and-Replace Method

This method often requires a significant investment of effort initially, but permits extremely flexible addressing. For details, see section *16.2.5. Canonical References*.

<cRefPattern> (canonical reference pattern) specifies an expression and replacement pattern for transforming a canonical reference into a URI.

2.3.5.3 Milestone Method

This method is appropriate when only 'milestone' tags (see section *3.10.3. Milestone Elements*) are available to provide the required referencing information. It does not provide any abilities which cannot be mimicked by the search-and-replace referencing method discussed in the previous section, but in the cases where it applies, it provides a somewhat simpler notation.

A reference based on milestone tags concatenates the values specified by one or more such tags. Since each tag marks the point at which a value changes, it may be regarded as specifying the refState of a variable. A reference declaration using this method therefore specifies the individual components of the canonical reference as a sequence of <refState> elements:

<refState/> (reference state) specifies one component of a canonical reference defined by the milestone method.

@**ed** (edition) indicates which edition or version the milestone applies to.

@**unit** indicates what kind of state is changing at this milestone.

@**delim** (delimiter) supplies a delimiting string following the reference component.

@**length** specifies the fixed length of the reference component.

For example, the reference 'Matthew 12:34' might be thought of as representing the state of three variables: the book variable is in state 'Matthew'; the chapter variable is in state '12', and the verse variable is in state '34'. If milestone tagging has been used, there should be a tag marking the point in the text at which each of the above 'variables' changes its state.[5] To find 'Matthew 12:34' therefore an application must scan left to right through the text, monitoring changes in the state of each of these three variables as it does so. When all three are simultaneously in the required state, the desired point will have been reached. There may of course be several such points.

The **delim** and **length** attributes are used to specify components of a canonical reference using this method in exactly the same way as for the stepwise method described in the preceding section. The other attributes are

[5] On the <milestone> tag itself, what are here referred to as 'variables' are identified by the combination of the **ed** and **unit** attributes.

used to determine which instances of <milestone> tags in the text are to be checked for state-changes. A state-change is signalled whenever a new <milestone> tag is found with **unit** and, optionally, **ed** attributes identical to those of the <refState> element in question. The value for the new state may be given explicitly by the **n** attribute on the <milestone> element, or it may be implied, if the **n** attribute is not specified.

For example, for canonical references in the form *xx.yyy* where the *xx* represents the page number in the first edition, and *yyy* the line number within this page, a reference system declaration such as the following would be appropriate:

```
<refsDecl>
  <refState
     ed="first"
     unit="page"
     length="2"
     delim="."/>
  <refState ed="first" unit="line" length="3"/>
</refsDecl>
```

This implies that milestone tags of the form

```
<milestone n="II" ed="first" unit="page"/>
<milestone ed="first" unit="line"/>
```

will be found throughout the text, marking the positions at which page and line numbers change. Note that no value has been specified for the **n** attribute on the second milestone tag above; this implies that its value at each state change is monotonically increased. For more detail on the use of milestone tags, see section *3.10.3. Milestone Elements*.

The milestone referencing scheme, though conceptually simple, is not supported by a generic SGML or XML parser. Its use places a correspondingly greater burden of verification and accuracy on the encoder.

A reference system declaration which applies to more than one text or division of a text need not be repeated in the header of each such text. Instead, the **decls** attribute of each text (or subdivision of the text) to which the declaration applies may be used to supply a cross-reference to it, as further described in section *15.3. Associating Contextual Information with a Text*.

2.3.6 The Classification Declaration

The <classDecl> element is used to group together definitions or sources for any descriptive classification schemes used by other parts of the header. Each such scheme is represented by a <taxonomy> element, which may contain either a simple bibliographic citation, or a definition of the descriptive typology concerned; the following elements are used in defining a descriptive classification scheme:

<classDecl> (classification declarations) contains one or more taxonomies defining any classificatory codes used elsewhere in the text.

<taxonomy> defines a typology used to classify texts either implicitly, by means of a bibliographic citation, or explicitly by a structured taxonomy.

<category> contains an individual descriptive category, possibly nested within a superordinate category, within a user-defined taxonomy.

<catDesc> (category description) describes some category within a taxonomy or text typology, either in the form of a brief prose description or in terms of the situational parameters used by the TEI formal textDesc.

The <taxonomy> element has two slightly different, but related, functions. For well-recognized and documented public classification schemes, such as Dewey or other published descriptive thesauri, it contains simply a bibliographic citation indicating where a full description of a particular taxonomy may be found.

```
<taxonomy xml:id="ddc12">
  <bibl>
    <title>Dewey Decimal Classification</title>
    <edition>Abridged Edition 12</edition>
  </bibl>
</taxonomy>
```

For less easily accessible schemes, the <taxonomy> element contains a description of the taxonomy itself as well as an optional bibliographic citation. The description consists of a number of <category> elements, each defining a single category within the given typology. The category is defined by the contents of a nested <catDesc> element, which may contain either a phrase describing the category, or any number of elements from the model.catDescPart class. When the corpus module is included in a schema, this class provides the <textDesc> element whose components allow the definition of a text type in terms of a set of 'situational parameters' (see further section *15.2.1. The Text Description*; if the corpus module is not included in a schema, this class is empty and the <catDesc> element may contain only plain text.

If the category is subdivided, each subdivision is represented by a nested <category> element, having the same structure. Categories may be nested to an arbitrary depth in order to reflect the hierarchical structure of the taxonomy. Each <category> element bears a unique **xml:id** attribute, which is used as the target for <catRef> elements referring to it.

```
<taxonomy xml:id="b">
  <bibl>Brown Corpus</bibl>
  <category xml:id="b.a">
    <catDesc>Press Reportage</catDesc>
    <category xml:id="b.a1">
      <catDesc>Daily</catDesc>
    </category>
    <category xml:id="b.a2">
      <catDesc>Sunday</catDesc>
    </category>
    <category xml:id="b.a3">
      <catDesc>National</catDesc>
    </category>
    <category xml:id="b.a4">
      <catDesc>Provincial</catDesc>
    </category>
    <category xml:id="b.a5">
      <catDesc>Political</catDesc>
    </category>
    <category xml:id="b.a6">
      <catDesc>Sports</catDesc>
    </category>
  </category>
  <category xml:id="b.d">
    <catDesc>Religion</catDesc>
    <category xml:id="b.d1">
      <catDesc>Books</catDesc>
    </category>
    <category xml:id="b.d2">
```

```
     <catDesc>Periodicals and tracts</catDesc>
   </category>
  </category>
</taxonomy>
```

Linkage between a particular text and a category within such a taxonomy is made by means of the <catRef> element within the <textClass> element, as described in section *2.4.3. The Text Classification*. Where the taxonomy permits of classification along more than one dimension, more than one category will be referenced by a particular <catRef>, as in the following example, which identifies a text with the sub-categories 'Daily', 'National', and 'Political' within the category 'Press Reportage' as defined above.

```
<catRef  target="#b.a1 #b.a3 #b.a5"/>
```

2.3.7 The Application Information Element

It is sometimes convenient to store information relating to the processing of an encoded resource within its header. Typical uses for such information might be:

- to allow an application to discover that it has previously opened or edited a file, and what version of itself was used to do that;
- to show (through a date) which application last edited the file to allow for diagnosis of any problems that might have been caused by that application;
- to allow users to discover information about an application used to edit the file
- to allow the application to declare an interest in elements of the file which it has edited, so that other applications or human editors may be more wary of making changes to those sections of the file.

The class model.applicationLike provides an element, <application>, which may be used to record such information within the <appInfo> element.

<appInfo> (application information) records information about an application which has edited the TEI file.

<application> provides information about an application which has acted upon the document.

@**ident** Supplies an identifier for the application, independent of its version number or display name.

@**version** Supplies a version number for the application, independent of its identifier or display name.

Each <application> element identifies the current state of one software application with regard to the current file. This element is a member of the att.datable class, which provides a variety of attributes for associating this state with a date and time, or a temporal range. The **ident** and **version** attributes should be used to uniquely identify the application and its major version number (for example, **ImageMarkupTool 1.5**). It is not intended that an application should add a new <application> each time it touches the file.

The following example shows how these elements might be used to document the fact that version 1.5 of an application called 'Image Markup Tool' has an interest in two parts of a document which was last saved on June 6 2006. The parts concerned are accessible at the URLs given as target for the two <ptr> elements.

```
<appInfo>
 <application  version="1.5"  ident="ImageMarkupTool"  notAfter="2006-06-01">
  <label>Image Markup Tool</label>
```

```
    <ptr target="#P1"/>
    <ptr target="#P2"/>
  </application>
</appInfo>
```

2.3.8 Module-Specific Declarations

The elements discussed so far are available to any schema. When the schema in use includes some of the more specialised TEI modules, these make available other more module-specific components of the encoding declaration. These are discussed fully in the documentation for the module in question, but are also noted briefly here for convenience.

The <fsdDecl> element is available only when the iso-fs module is included in a schema. Its purpose is to document the feature system declaration (as defined in chapter *18.11. Feature System Declaration*) underlying any analytic feature structures (as defined in chapter *18. Feature Structures*) present in the text documented by this header.

The <metDecl> element is available only when the verse module is included in a schema. Its purpose is to document any metrical notation scheme used in the text, as further discussed in section *6.3. Rhyme and Metrical Analysis*. It consists either of a prose description or a series of <metSym> elements.

The <variantEncoding> element is available only when the textcrit module is included in a schema. Its purpose is to document the method used to encode textual variants in the text, as discussed in section *12.2. Linking the Apparatus to the Text*.

2.4 The Profile Description

The <profileDesc> element is the third major subdivision of the TEI Header. It is an optional element, the purpose of which is to enable information characterizing various descriptive aspects of a text or a corpus to be recorded within a single unified framework.

<profileDesc> (text-profile description) provides a detailed description of non-bibliographic aspects of a text, specifically the languages and sublanguages used, the situation in which it was produced, the participants and their setting.

In principle, almost any component of the header might be of importance as a means of characterizing a text. The author of a written text, its title or its date of publication, may all be regarded as characterizing it at least as strongly as any of the parameters discussed in this section. The rule of thumb applied has been to exclude from discussion here most of the information which generally forms part of a standard bibliographic style description, if only because such information has already been included elsewhere in the TEI header.

The core <profileDesc> element has three optional components, represented by the following elements:

<creation> contains information about the creation of a text.

<langUsage> (language usage) describes the languages, sublanguages, registers, dialects, etc. represented within a text.

<textClass> (text classification) groups information which describes the nature or topic of a text in terms of a standard classification scheme, thesaurus, etc.

These elements are further described in the remainder of this section.

Three other elements may also appear within the <profileDesc> element when the corpus module described in chapter *15. Language Corpora* is included in a schema:

<textDesc> (text description) provides a description of a text in terms of its situational parameters.

<particDesc> (participation description) describes the identifiable speakers, voices, or other participants in a linguistic interaction.

<settingDesc> (setting description) describes the setting or settings within which a language interaction takes place, either as a prose description or as a series of setting elements.

For descriptions of these elements, see section *15.2. Contextual Information*.

The following element can appear in the <profileDesc> element when the transcr module for the transcription of primary sources described in chapter *11. Representation of Primary Sources* is included in a schema:

<handNotes> contains one or more <handNote> elements documenting the different hands identified within the source texts.

For a description of this element, see section *11.4.1. Document Hands*. Its purpose is to group together a number of <handNote> elements, each of which describes a different hand or equivalent identified within a manuscript. The <handNote> element can also appear within a structured manuscript description, when the msdescription module described in chapter *10. Manuscript Description* is included in a schema. For this reason, the <handNote> element is actually declared within the header module, but is only accessible to a schema when one or other of the transcr or msdescription modules is included in a schema. See further the discussion at *11.4.1. Document Hands*.

2.4.1 Creation

The <creation> element contains phrases describing the origin of the text, e.g. the date and place of its composition.

<creation> contains information about the creation of a text.

The date and place of composition are often of particular importance for studies of linguistic variation; since such information cannot be inferred with confidence from the bibliographic description of the copy text, the <creation> element may be used to provide a consistent location for this information:

```
<creation>
  <date when="1992-08">August 1992</date>
  <rs type="city">Taos, New Mexico</rs>
</creation>
```

2.4.2 Language Usage

The <langUsage> element is used within the <profileDesc> element to describe the languages, sublanguages, registers, dialects, etc. represented within a text. It contains one or more <language> elements, each of which provides information about a single language, notably the quantity of that language present in the text. Note that this element should *not* be used to supply information about any non-standard characters or glyphs used by this language; such information should be recorded in the <charDecl> element in the encoding description (see further *5. Representation of Non-standard Characters and Glyphs*).

<langUsage> (language usage) describes the languages, sublanguages, registers, dialects, etc. represented within a text.

<language> characterizes a single language or sublanguage used within a text.

- @**usage** specifies the approximate percentage (by volume) of the text which uses this language.
- @**ident** (identifier) Supplies a language code constructed as defined in BCP 47 which is used to identify the language documented by this element, and which is referenced by the global **xml:lang** attribute.

A <language> element may be supplied for each different language used in a document. If used, its **ident** attribute should specify an appropriate language identifier, as further discussed in section *vi.1 Language*

identification. This is particularly important if extended language identifiers have been used as the value of **xml:lang** attributes elsewhere in the document.

Here is an example of the use of this element:

```
<langUsage>
  <language ident="fr-CA" usage="60">Québecois</language>
  <language ident="en-CA" usage="20">Canadian business English</language>
  <language ident="en-GB" usage="20">British English</language>
</langUsage>
```

2.4.3 The Text Classification

The second component of the core <profileDesc> element is the <textClass> element. This element is used to classify a text according to one or more of the following methods:

- by reference to a recognized international classification such as the Dewey Decimal Classification, the Universal Decimal Classification, the Colon Classification, the Library of Congress Classification, or any other system widely used in library and documentation work
- by providing a set of keywords, as provided for example by British Library or Library of Congress Cataloguing in Publication data
- by referencing any other taxonomy of text categories recognized in the field concerned, or peculiar to the material in hand; this may include one based on recurring sets of values for the situational parameters defined in section *15.2.1. The Text Description*, or the demographic elements described in section *15.2.2. The Participant Description*

The last of these may be particularly important for dealing with existing corpora or collections, both as a means of avoiding the expense or inconvenience of reclassification and as a means of documenting the organizing principles of such materials.

The following elements are provided for this purpose:

<keywords> contains a list of keywords or phrases identifying the topic or nature of a text.
  @**scheme** identifies the controlled vocabulary within which the set of keywords concerned is defined.

<classCode> (classification code) contains the classification code used for this text in some standard classification system.
  @**scheme** identifies the classification system or taxonomy in use.

<catRef/> (category reference) specifies one or more defined categories within some taxonomy or text typology.
  @**target** identifies the categories concerned

The <keywords> element simply categorizes an individual text by supplying a list of keywords which may describe its topic or subject matter, its form, date, etc. In some schemes, the order of items in the list is significant, for example, from major topic to minor; in others, the list has an organized substructure of its own. No recommendations are made here as to which method is to be preferred. Wherever possible, such keywords should be taken from a recognized source, such as the British Library/Library of Congress Cataloguing in Publication data in the case of printed books, or a published thesaurus appropriate to the field.

The **scheme** attribute should be used to indicate the source of the keywords used. This is done by supplying the value used for the **xml:id** attribute of a <taxonomy> element within which further details of the source concerned may be found. The <taxonomy> element occurs in the <classDecl> part of the encoding declarations within the TEI Header and is described in section *2.3.6. The Classification Declaration*. For example:

```
<keywords  scheme="#lcsh">
  <list>
    <item>Data base management</item>
    <item>SQL (Computer program language)</item>
  </list>
</keywords>
```

```
<keywords  scheme="#lcsh">
  <list>
    <item>English literature -- History and criticism -- Data processing.</item>
    <item>English literature -- History and criticism -- Theory, etc.</item>
    <item>English language -- Style -- Data processing.</item>
    <item>Style, Literary -- Data processing.</item>
  </list>
</keywords>
```

The <classCode> element also categorizes an individual text, by supplying a numerical or other code used in a recognized classification scheme, such as the Dewey Decimal Classification. The **scheme** attribute is used to indicate the source of the classification scheme: this may be a pointer of any kind, either to a TEI element, likely in the current document, as in the <keywords> examples above, or to some canonical source for the scheme, as in the following example:

```
<classCode  scheme="http://www.example.com/udc">005.756</classCode>
<classCode  scheme="#lc">QA76.9</classCode>
```

```
<classCode  scheme="http://www.example.com/udc">820.285</classCode>
```

The <catRef> element categorizes an individual text by pointing to one or more <category> elements. The <category> element (which is fully described in section *2.3.6. The Classification Declaration*) holds information about a particular classification or category within a given taxonomy. Each such category must have a unique identifier, which may be supplied as the value of the **target** attribute for <catRef> elements which are regarded as falling within the category indicated.

A text may, of course, fall into more than one category, in which case more than one identifier will be supplied as the value for the **target** attribute on the <catRef> element, as in the following example:

```
<catRef  target="#b.a4 #b.d2"/>
```

The **scheme** attribute may be supplied to specify the taxonomy to which the categories identified by the target attribute belong, if this is not adequately conveyed by the resource pointed to. For example,

```
<catRef
   target="#b.a4 #b.d2"
   scheme="http://www.example.com/browncorpus"/>
<catRef  target="http://www.example.com/SUC/#A45"/>
```

Here the same text has been classified as of categories **b.a4** and **b.d2** within the Brown classification scheme (presumed to be available from http://www.example.com/browncorpus), and as of category 'A45' within the SUC classification scheme documented at the URL given.

The distinction between the <catRef> and <classCode> elements is that the values used as identifying codes are exhaustively enumerated, typically with the header, for the former, while the latter may be used to indicate a more open ended or descriptive classification system.

2.5 The Revision Description

The final sub-element of the TEI header, the <revisionDesc> element, provides a detailed change log in which each change made to a text may be recorded. Its use is optional but highly recommended. It provides essential information for the administration of large numbers of files which are being updated, corrected, or otherwise modified as well as extremely useful documentation for files being passed from researcher to researcher or system to system. Without change logs, it is easy to confuse different versions of a file, or to remain unaware of small but important changes made in the file by some earlier link in the chain of distribution. No change should be made in any TEI-conformant file without corresponding entries being made in the change log.

<revisionDesc> (revision description) summarizes the revision history for a file.

<change> summarizes a particular change or correction made to a particular version of an electronic text which is shared between several researchers.

The main purpose of the revision description is to record changes in the text to which a header is prefixed. However, it is recommended TEI practice to include entries also for significant changes in the header itself (other than the revision description itself, of course). At the very least, an entry should be supplied indicating the date of creation of the header.

The log consists of a list of entries, one for each change. This may be encoded using either the regular <list> element, as described in section *3.7. Lists* or as a series of special purpose <change> elements, each of which contains a more detailed description of the changes made. The attributes **when** and **who** are used to indicate the date of the change and the person responsible for it respectively. The description of the change itself can range from a simple phrase to a series of paragraphs. If a number is to be associated with one or more changes (for example, a revision number), the global **n** attribute may be used to indicate it.

It is recommended to give changes in reverse chronological order, most recent first.

For example:

```
<titleStmt>
  <title>The Amorous Prince, or, the Curious Husband, 1671</title>
  <author>
    <persName  ref="#abehn.aeh">Behn, Aphra</persName>
  </author>
  <respStmt  xml:id="pcaton.xzc">
    <persName>Caton, Paul</persName>
    <resp>electronic publication editor</resp>
  </respStmt>
  <respStmt  xml:id="wgui.ner">
    <persName>Gui, Weihsin</persName>
    <resp>encoder</resp>
  </respStmt>
  <respStmt  xml:id="jwernimo.lrv">
    <persName>Wernimont, Jacqueline</persName>
    <resp>encoder</resp>
  </respStmt>
</titleStmt>
<!-- ... -->
```

```
<revisionDesc>
  <change  n="RCS:1.39"  when="2007-08-08"  who="#jwernimo.lrv">Changed <val>drama.verse</val>
    <gi>lg</gi>s to <gi>p</gi>s. <note>we have opened a discussion about the need for a new
        value for <att>type</att> of <gi>lg</gi>, <val>drama.free.verse</val>, in order to address
        the verse of Behn which is not in regular iambic pentameter. For the time being these
        instances are marked with a comment note until we are able to fully consider the best way
        to encode these instances.</note>
  </change>
  <change  n="RCS:1.33"  when="2007-06-28"  who="#pcaton.xzc">Added <att>key</att> and <att>reg</att>
     to <gi>name</gi>s.</change>
  <change  n="RCS:1.31"  when="2006-12-04"  who="#wgui.ner">Completed renovation.  Validated.</change>
</revisionDesc>
```

In the above example, the **who** attributes point to <respStmt> elements; they could equally well point to <person> elements.

2.6 Minimal and Recommended Headers

The TEI header allows for the provision of a very large amount of information concerning the text itself, its source, its encodings, and revisions of it, as well as a wealth of descriptive information such as the languages it uses and the situation(s) in which it was produced, together with the setting and identity of participants within it. This diversity and richness reflects the diversity of uses to which it is envisaged that electronic texts conforming to these Guidelines will be put. It is emphatically *not* intended that all of the elements described above should be present in every TEI Header.

The amount of encoding in a header will depend both on the nature and the intended use of the text. At one extreme, an encoder may expect that the header will be needed only to provide a bibliographic identification of the text adequate to local needs. At the other, wishing to ensure that their texts can be used for the widest range of applications, encoders will want to document as explicitly as possible both bibliographic and descriptive information, in such a way that no prior or ancillary knowledge about the text is needed in order to process it. The header in such a case will be very full, approximating to the kind of documentation often supplied in the form of a manual. Most texts will lie somewhere between these extremes; textual corpora in particular will tend more to the latter extreme. In the remainder of this section we demonstrate first the minimal, and next a commonly recommended, level of encoding for the bibliographic information held by the TEI header.

Supplying only the minimal level of encoding required, the TEI header of a single text might look like the following example:

```
<teiHeader>
  <fileDesc>
    <titleStmt>
      <title>Thomas Paine:  Common sense, a
            machine-readable transcript</title>
      <respStmt>
        <resp>compiled by</resp>
        <name>Jon K Adams</name>
      </respStmt>
    </titleStmt>
    <publicationStmt>
      <distributor>Oxford Text Archive</distributor>
    </publicationStmt>
    <sourceDesc>
      <bibl>The complete writings of Thomas Paine, collected and edited
            by Phillip S. Foner (New York, Citadel Press, 1945)</bibl>
```

```
   </sourceDesc>
  </fileDesc>
</teiHeader>
```

The only mandatory component of the TEI Header is the <fileDesc> element. Within this, <titleStmt>, <publicationStmt>, and <sourceDesc> are all required constituents. Within the title statement, a title is required, and an author should be specified, even if it is *unknown*, as should some additional statement of responsibility, here given by the <respStmt> element. Within the <publicationStmt>, a publisher, distributor, or other agency responsible for the file must be specified. Finally, the source description should contain at the least a loosely structured bibliographic citation identifying the source of the electronic text if (as is usually the case) there is one.

We now present the same example header, expanded to include additionally recommended information, adequate to most bibliographic purposes, in particular to allow for the creation of an AACR2-conformant bibliographic record. We have also added information about the encoding principles used in this (imaginary) encoding, about the text itself (in the form of Library of Congress subject headings), and about the revision of the file.

```
<teiHeader>
  <fileDesc>
    <titleStmt>
      <title>Common sense, a machine-readable transcript</title>
      <author>Paine, Thomas (1737-1809)</author>
      <respStmt>
        <resp>compiled by</resp>
        <name>Jon K Adams</name>
      </respStmt>
    </titleStmt>
    <editionStmt>
      <edition>
        <date>1986</date>
      </edition>
    </editionStmt>
    <publicationStmt>
      <distributor>Oxford Text Archive.</distributor>
      <address>
        <addrLine>Oxford University Computing Services,</addrLine>
        <addrLine>13 Banbury Road,</addrLine>
        <addrLine>Oxford OX2 6RB,</addrLine>
        <addrLine>UK</addrLine>
      </address>
    </publicationStmt>
    <notesStmt>
      <note>Brief notes on the text are in a
            supplementary file.</note>
    </notesStmt>
    <sourceDesc>
      <biblStruct>
        <monogr>
          <editor>Foner, Philip S.</editor>
          <title>The collected writings of Thomas Paine</title>
          <imprint>
            <pubPlace>New York</pubPlace>
            <publisher>Citadel Press</publisher>
```

```
        <date>1945</date>
       </imprint>
      </monogr>
    </biblStruct>
  </sourceDesc>
</fileDesc>
<encodingDesc>
  <samplingDecl>
    <p>Editorial notes in the Foner edition have not
         been reproduced. </p>
    <p>Blank lines and multiple blank spaces, including paragraph
         indents, have not been preserved. </p>
  </samplingDecl>
  <editorialDecl>
    <correction  status="high"  method="silent">
      <p>The following errors
             in the Foner edition have been corrected:
      <list>
         <item>p. 13 l. 7 cotemporaries contemporaries </item>
         <item>p. 28 l. 26 [comma] [period] </item>
         <item>p. 84 l. 4 kin kind </item>
         <item>p. 95 l. 1 stuggle struggle </item>
         <item>p. 101 l. 4 certainy certainty </item>
         <item>p. 167 l. 6 than that </item>
         <item>p. 209 l. 24 publshed published </item>
       </list>
      </p>
    </correction>
    <normalization>
      <p>No normalization beyond that performed
             by Foner, if any. </p>
    </normalization>
    <quotation  marks="all"  form="std">
      <p>All double quotation marks
             rendered with ", all single quotation marks with
             apostrophe. </p>
    </quotation>
    <hyphenation  eol="none">
      <p>Hyphenated words that appear at the
             end of the line in the Foner edition have been reformed.</p>
    </hyphenation>
    <stdVals>
      <p>The values of <att>when-iso</att> on the <gi>time</gi>
             element always end in the format <val>HH:MM</val> or
      <val>HH</val>; i.e., seconds, fractions thereof, and time
             zone designators are not present.</p>
    </stdVals>
    <interpretation>
      <p>Compound proper names are marked. </p>
      <p>Dates are marked. </p>
      <p>Italics are recorded without interpretation. </p>
    </interpretation>
  </editorialDecl>
  <classDecl>
    <taxonomy  xml:id="lcsh">
      <bibl>Library of Congress Subject Headings</bibl>
    </taxonomy>
```

```
    <taxonomy xml:id="lc">
      <bibl>Library of Congress Classification</bibl>
    </taxonomy>
  </classDecl>
 </encodingDesc>
 <profileDesc>
  <creation>
    <date>1774</date>
  </creation>
  <langUsage>
    <language ident="en" usage="100">English.</language>
  </langUsage>
  <textClass>
    <keywords scheme="#lcsh">
      <list>
        <item>Political science</item>
        <item>United States -- Politics and government —
                Revolution, 1775-1783</item>
      </list>
    </keywords>
    <classCode scheme="#lc">JC 177</classCode>
  </textClass>
 </profileDesc>
 <revisionDesc>
  <change when="1996-01-22">
    <name>CMSMcQ</name> finished proofreading
  </change>
  <change when="1995-10-30">
    <name>L.B. </name> finished proofreading
  </change>
  <change when="1995-07-20">
    <name>R.G. </name> finished proofreading
  </change>
  <change when="1995-07-04">
    <name>R.G. </name> finished data entry
  </change>
  <change when="1995-01-15">
    <name>R.G. </name> began data entry
  </change>
 </revisionDesc>
</teiHeader>
```

Many other examples of recommended usage for the elements discussed in this chapter are provided here, in the reference index and in the associated tutorials.

2.7 Note for Library Cataloguers

A strong motivation in preparing the material in this chapter was to provide in the TEI file header a viable chief source of information for cataloguing computer files. The file header is not a library catalogue record, and so will not make all of the distinctions essential in standard library work. It also includes much information generally excluded from standard bibliographic descriptions. It is the intention of the developers, however, to ensure that the information required for a catalogue record be retrievable from the TEI file header, and moreover that the mapping from the one to the other be as simple and straightforward as possible. Where the correspondence is not obvious, it may prove useful to consult one of the works which were influential in developing the content of the TEI file header. These include:

ISBD(G) The International Standard Book Description (General) is an international standard setting out what information should be recorded in a description of a bibliographical item. There are also separate ISBDs covering different types of material, e.g. ISBD(M) for monographs, ISBD(ER) for electronic resources. These separate ISBDs follow the same general scheme as the main ISBD(G), but provide appropriate interpretations for the specific materials under consideration.

AACR2 The Anglo-American Cataloguing Rules, Second Edition, 2002 Revision: 2005 Update are the official guidelines for the construction of catalogues in general libraries in the English-speaking world. Other national cataloguing codes exist as well. AACR2 is explicitly based on the general framework of the ISBD(G) and the subsidiary ISBDs: it gives a description of how to catalogue items according to the ISBDs, and how to construct indexes and cross-references.

ANSI Z.39.29 ANSI Z.39.29 is an American national standard governing bibliographic references for use in bibliographies, end-of-work lists, references in abstracting and indexing publications, and outputs from computerized bibliographic data bases. This standard has however now been withdrawn, pending substantial revision. The international standard which covers the same area is ISO 690:1987. Other relevant standards include BS 1629:1989, BS 5605:1978, and BS 6371:1983.

2.8 The TEI Header Module

The module described in this chapter makes available the following components:

Module header: The TEI Header

- Elements defined: appInfo application authority availability biblFull cRefPattern catDesc catRef category change classCode classDecl correction creation distributor edition editionStmt editorialDecl encodingDesc extent fileDesc funder geoDecl handNote hyphenation idno interpretation keywords langUsage language namespace normalization notesStmt principal profileDesc projectDesc publicationStmt quotation refState refsDecl rendition revisionDesc samplingDecl segmentation seriesStmt sourceDesc sponsor stdVals tagUsage tagsDecl taxonomy teiHeader textClass titleStmt
- Classes defined: model.applicationLike model.editorialDeclPart model.encodingPart model.headerPart model.profileDescPart model.sourceDescPart

The selection and combination of modules to form a TEI schema is described in *1.2. Defining a TEI Schema*.

Chapter 3

Elements Available in All TEI Documents

This chapter describes elements which may appear in any kind of text and the tags used to mark them in all TEI documents. Most of these elements are freely floating phrases, which can appear at any point within the textual structure, although they must generally be contained by a higher-level element of some kind (such as a paragraph). A few of the elements described in this chapter (for example, bibliographic citations and lists) have a comparatively well-defined internal structure, but most of them have no consistent inner structure of their own. In the general case, they contain only a few words, and are often identifiable in a conventionally printed text by the use of typographic conventions such as shifts of font, use of quotation or other punctuation marks, or other changes in layout.

This chapter begins by describing the <p> tag used to mark paragraphs, the prototypical formal unit for running text in many TEI modules. This is followed, in section *3.2. Treatment of Punctuation*, by a discussion of some specific problems associated with the interpretation of conventional punctuation, and the methods proposed by the Guidelines for resolving ambiguities therein.

The next section (section *3.3. Highlighting and Quotation*) describes a number of phrase-level elements commonly marked by typographic features (and thus well-represented in conventional markup languages). These include features commonly marked by font shifts (section *3.3.2. Emphasis, Foreign Words, and Unusual Language*) and features commonly marked by quotation marks (section *3.3.3. Quotation*) as well as such features as terms, cited words, and glosses (section *3.3.4. Terms, Glosses, Equivalents, and Descriptions*).

Section *3.4. Simple Editorial Changes* introduces some phrase-level elements which may be used to record simple editorial interventions, such as emendation or correction of the encoded text. The elements described here constitute a simple subset of the full mechanisms for encoding such information (described in full in chapter *11. Representation of Primary Sources*), which should be adequate to most commonly encountered situations.

The next section (section *3.5. Names, Numbers, Dates, Abbreviations, and Addresses*) describes several phrase-level and inter-level elements which, although often of interest for analysis or processing, are rarely explicitly identified in conventional printing. These include names (section *3.5.1. Referring Strings*), numbers and measures (section *3.5.3. Numbers and Measures*), dates and times (section *3.5.4. Dates and Times*), abbreviations (section *3.5.5. Abbreviations and Their Expansions*), and addresses (section *3.5.2. Addresses*).

In the same way, the following section (section *3.6. Simple Links and Cross-References*) presents only a subset of the facilities available for the encoding of cross-references or text-linkage. The full story may be found in chapter *16. Linking, Segmentation, and Alignment*; the tags presented here are intended to be usable for a wide variety of simple applications.

Sections *3.7. Lists*, and *3.8. Notes, Annotation, and Indexing*, describe two kinds of quasi-structural elements: lists and notes. These may appear either within chunk-level elements such as paragraphs, or between them. Several kinds of lists are catered for, of an arbitrary complexity. The section on notes discusses both notes found

in the source and simple mechanisms for adding annotations of an interpretive nature during the encoding; again, only a subset of the facilities described in full elsewhere (specifically, in chapter *17. Simple Analytic Mechanisms*) is discussed.

Section *3.9. Graphics and other non-textual components* introduces some simple ways of representing graphic or other non-textual content found in a text. A fuller discussion of the multimedia facilities supported by these Guidelines may be found in chapters *14. Tables, Formulæ, and Graphics* and *16. Linking, Segmentation, and Alignment*.

Next, section *3.10. Reference Systems*, describes methods of encoding within a text the conventional system or systems used when making references to the text. Some reference systems have attained canonical authority and must be recorded to make the text useable in normal work; in other cases, a convenient reference system must be created by the creator or analyst of an electronic text.

Like lists and notes, the bibliographic citations discussed in section *3.11. Bibliographic Citations and References*, may be regarded as structural elements in their own right. A range of possibilities is presented for the encoding of bibliographic citations or references, which may be treated as simple phrases within a running text, or as highly-structured components suitable for inclusion in a bibliographic database.

Additional elements for the encoding of passages of verse or drama (whether prose or verse) are discussed in section *3.12. Passages of Verse or Drama*.

The chapter concludes with a technical overview of the structure and organization of the module described here. This should be read in conjunction with chapter *1. The TEI Infrastructure*, describing the structure of the TEI document type definition.

3.1 Paragraphs

The paragraph is the fundamental organizational unit for all prose texts, being the smallest regular unit into which prose can be divided. Prose can appear in all TEI texts, even those that are primarily of another genre (e.g., verse); thus the paragraph is described here, as an element which can appear in any kind of text.

Paragraphs can contain any of the other elements described within this chapter, as well as some other elements which are specific to individual text types. We distinguish phrase-level elements, which must be entirely contained within a paragraph and cannot appear except within one, from chunks, which can appear between, but not within, paragraphs, and from inter-level elements, which can appear either within a single paragraph or between paragraphs. The class of phrases includes emphasized or quoted phrases, names, dates, etc. The class of inter-level elements includes bibliographic citations, notes, lists, etc. The class of chunks includes the paragraph itself, and other elements which have similar structural properties, notably the <ab> (anonymous block) element described in *16.3. Blocks, Segments, and Anchors*) which may be used as an alternative to the paragraph in some kinds of texts.

Because paragraphs may appear in different base or additional tag sets, their possible contents may differ in different kinds of documents. In particular, additional elements not listed in this chapter may appear in paragraphs in certain kinds of text. However, the elements described in this chapter are always by default available in all kinds of text.

The paragraph is marked using the <p> element:

<p> (paragraph) marks paragraphs in prose.

If a consistent internal subdivision of paragraphs is desired, the <s> or <seg> ('segment') elements may be used, as discussed in chapters *16. Linking, Segmentation, and Alignment* and *17. Simple Analytic Mechanisms* respectively. More usually, however, paragraphs have no firm internal structure, but contain prose encoded as a mix of characters, entity references, phrases marked as described in the rest of this chapter, and embedded elements like lists, figures, or tables.

Since paragraphs are usually explicitly marked in Western texts, typically by indentation, the application of the <p> tag usually presents few problems.

In some cases, the body of a text may comprise but a single paragraph:

```
<body>
  <p>I fully appreciate Gen. Pope's splendid achievements with their
     invaluable results; but you must know that Major Generalships in the
     Regular Army, are not as plenty as blackberries.</p>
</body>
```

Source: [128]

This news story shows typically short journalistic paragraphs:

```
<head>SARAJEVO, Bosnia and Herzegovina, April 19</head>
<p>Serbs seized more territory in this struggling new country today as
the United States Air Force ended a two-day airlift of humanitarian
aid into the capital, Sarajevo.</p>
<p>International relief workers called on European Community nations
to step up their humanitarian aid to the former Yugoslav republic,
in conjunction with new American aid flights if necessary.</p>
<p>A special envoy from the European Community, Colin Doyle, harshly
condemned the decision by Serbs to shell Sarajevo on Saturday night
during a visit to the Bosnian capital by a senior American official,
Deputy Assistant Secretary of State Ralph R. Johnson.</p>
<p>...</p>
```

The following extract from a Russian fairy tale demonstrates how other phrase level elements (in this case <q> elements representing direct speech; see section *3.3.3. Quotation*) may be nested within, but not across, paragraphs:

```
<p>A fly built a castle, a tall and mighty castle.
There came to the castle the Crawling Louse. <q>Who,
    who's in the castle? Who, who's in your house?</q>
said the Crawling Louse. <q>I, I, the Languishing Fly.
    And who art thou?</q>
  <q>I'm the Crawling Louse.</q>
</p>
<p>Then came to the castle the Leaping Flea. <q>Who,
    who's in the castle?</q> said the Leaping Flea. <q>I,
    I, the Languishing Fly, and I, the Crawling Louse. And
    who art thou?</q>
  <q>I'm the Leaping Flea.</q>
</p>
<p>Then came to the castle the Mischievous Mosquito.
<q>Who, who's in the castle?</q> said the Mischievous
Mosquito. <q>I, I, the Languishing Fly, and I, the
    Crawling Louse, and I, the Leaping Flea. And who art
    thou?</q>
  <q>I'm the Mischievous Mosquito.</q>
</p>
```

Source: [31]

3.2 Treatment of Punctuation

Punctuation marks cause problems for text markup when they are not available in the character set used and when they are significantly ambiguous. To a large extent, the availability of the Unicode character set addresses

most such problems, since it provides specific code points for most punctuation marks, and also distinguishes glyphs (such as stop, comma, and hyphen) which are used with different functions. Thus, for example, different Unicode code points are available for the hyphen used as a minus sign, as a word breaking hyphen, as a soft hyphen, or as a 'non-breaking' hyphen. The facilities described in chapter *5. Representation of Non-standard Characters and Glyphs* may also be used to define markup for non-standard punctuation characters.

Full stop (period) may mark (orthographic) sentence boundaries, abbreviations, decimal points, or serve as a visual aid in printing numbers. These usages can be distinguished by tagging S-units, abbreviations, and numbers, as described in sections *16.3. Blocks, Segments, and Anchors*, *3.5.5. Abbreviations and Their Expansions*, and *3.5.3. Numbers and Measures*. However, there are independent reasons for tagging these, whether or not they are marked by full stops, and the polysemy of the full stop itself is perhaps no different from that of any character in the writing system.

Question mark and exclamation mark typically mark the end of orthographic sentences, but may also be used as a mid-sentence comment by the author (*!* to express surprise or some other strong feeling, *?* to query a word or expression or mark a sentence as dubious in linguistic discussion). These uses may be distinguished by marking S-units, in which case the mid-sentence uses of these punctuation marks may be left unmarked, or tagged using the <c> element discussed in *17.1. Linguistic Segment Categories*.

Dashes are used for a variety of purposes: insertion, interruption, new speaker (in dialogue), list item. In the latter two cases it is preferable to mark the underlying feature using the elements <q> or <item>, on which see section *3.3.3. Quotation*, and section *3.7. Lists*, respectively.

Quotation marks may be removed from text contained by <q> or <quote> elements, especially as quotations are not always marked by quotation marks (notably long quotations) or may be marked in a variety of ways; see the discussion of quotation and related features in section *3.3.3. Quotation*.

Apostrophes must be distinguished from single quote marks. As with hyphens, this disambiguation may be performed by selecting an appropriate Unicode character, but it may also be represented by using explicit XML tags for quotations as suggested above. However, apostrophes have a variety of uses. In English they mark contractions, genitive forms, and (occasionally) plural forms. Full disambiguation of these uses belongs to the level of linguistic analysis and interpretation.

Parentheses and other marks of suspension such as dashes or ellipses are often used to signal information about the syntactic structure of a text fragment. Full disambiguation of their uses also belongs to the level of linguistic analysis and interpretation, and is therefore discussed in chapter *17. Simple Analytic Mechanisms*.

Where punctuation marks are disambiguated by tagging the underlying feature they signal, it may be debated whether they should be excluded or left as part of the text. In the case of quotation marks, it may be more convenient to distinguish opening from closing marks simply by using the appropriate Unicode character than to use the <q> element, with or without a **rend** attribute. The solution chosen will vary depending upon the feature and depending upon the purpose of the project.

3.3 Highlighting and Quotation

This section deals with a variety of textual features, all of which have in common that they are frequently realized in conventional printing practice by the use of such features as underlining, italic fonts, or quotation marks, collectively referred to here as highlighting. After an initial discussion of this phenomenon and alternate approaches to encoding it, this section describes ways of encoding the following textual features, all of which are conventionally rendered using some kind of highlighting:

- emphasis, foreign words and other linguistically distinct uses of highlighting
- representation of speech and thought, quotation, etc.
- technical terms, glosses, etc.

3.3.1 What Is Highlighting?

By *highlighting* we mean the use of any combination of typographic features (font, size, hue, etc.) in a printed or written text in order to distinguish some passage of a text from its surroundings.[1] The purpose of highlighting is generally to draw the reader's attention to some feature or characteristic of the passage highlighted; this section describes the elements recommended by these Guidelines for the encoding of such textual features.

In conventionally printed modern texts, highlighting is often employed to identify words or phrases which are regarded as being one or more of the following:

- distinct in some way — as foreign, dialectal, archaic, technical, etc.
- emphatic, and which would for example be stressed when spoken
- not part of the body of the text, for example cross-references, titles, headings, labels, etc.
- identified with a distinct narrative stream, for example an internal monologue or commentary.
- attributed by the narrator to some other agency, either within the text or outside it: for example, direct speech or quotation.
- set apart from the text in some other way: for example, proverbial phrases, words mentioned but not used, names of persons and places in older texts, editorial corrections or additions, etc.

The textual functions indicated by highlighting may not be rendered consistently in different parts of a text or in different texts. (For example, a foreign word may appear in italics if the surrounding text is in roman, but in roman if the surrounding text is in italics.) For this reason, these Guidelines distinguish between the encoding of rendering itself and the encoding of the underlying feature expressed by it.

Highlighting as such may be encoded by using either of the global attributes **rend** or **rendition** attributes (see *1.3.1.1. Global Attributes*). This allows the encoder both to specify the function of a highlighted phrase or word, by selecting the appropriate element described here or elsewhere in the Guidelines, and to further describe the way in which it is highlighted, by means of the **rend** attribute. If the encoder wishes to offer no interpretation of the feature underlying the use of highlighting in the source text, then the <hi> element may be used, which indicates only that the text so tagged was highlighted in some way.

<hi> (highlighted) marks a word or phrase as graphically distinct from the surrounding text, for reasons concerning which no claim is made.

The <hi> element is provided by the model.hiLike class.

The possible values carried by the **rend** attribute are not formally defined in this version of the Guidelines. Since the **rend** attribute may be used to document any peculiarity of the way a given segment of text was rendered in the original source text, it may need to express a very large range of typographic features, by no means restricted to typeface, type size, etc.

Where it is both appropriate and feasible, these Guidelines recommend that the textual feature marked by the highlighting should be encoded, rather than just the simple fact of the highlighting. This is for the following reasons:

- the same kind of highlighting may be used for different purposes in different contexts
- the same textual function may be highlighted in different ways in different contexts
- for analytic purposes, it is in general more useful to know the intended function of a highlighted phrase than simply that it is distinct.

In many, if not most, cases the underlying function of a highlighted phrase will be obvious and non-controversial, since the distinctions indicated by a change of highlighting correspond with distinctions discussed elsewhere in these Guidelines. The elements available to record such distinctions are, for the most

[1] Although the way in which a spoken text is performed, (for example, the voice quality, loudness, etc.) might be regarded as analogous to 'highlighting' in this sense, these Guidelines recommend distinct elements for the encoding of such 'highlighting' in spoken texts. See further section *8.3.6. Shifts*.

part, members of the model.emphLike class. This and the model.hiLike class mentioned above constitute the model.highlighted class, which is a phrase level class. Members of this class may appear anywhere within paragraph level elements.

The distinction between the two classes is simple, and typified by the two elements <hi> and <emph>: the former marks simply that a passage is typographically distinct in some way, while the latter asserts that a passage is linguistically emphasized for some purpose. These two properties, though often combined, are not identical. It should however be recognized, however, that cases do exist in which it is not economically feasible to mark the underlying function (e.g. in the preparation of large text corpora), as well as cases in which it is not intellectually appropriate (as in the transcription of some older materials, or in the preparation of material for the study of typographic practice). In such cases, the <hi> element or some other element from the model.hiLike class should be used.

Elements which are sometimes realized by typographic distinction but which are not discussed in this section include <title> (discussed in section *3.11. Bibliographic Citations and References*) and <name> (discussed in section *3.5.1. Referring Strings*).

3.3.2 Emphasis, Foreign Words, and Unusual Language

This subsection discusses the following elements:

<foreign> (foreign) identifies a word or phrase as belonging to some language other than that of the surrounding text.

<emph> (emphasized) marks words or phrases which are stressed or emphasized for linguistic or rhetorical effect.

<distinct> identifies any word or phrase which is regarded as linguistically distinct, for example as archaic, technical, dialectal, non-preferred, etc., or as forming part of a sublanguage.

These elements are all members of the model.emphLike class.

3.3.2.1 Foreign Words or Expressions

Words or phrases which are not in the main language of the text should be tagged as such, at least where the fact is indicated in the text. Where the word or phrase concerned is already distinguished from the rest of the text by virtue of its function (for example, because it is a name, a technical term, a quotation, a mentioned word, etc.) then the global **xml:lang** attribute should be used to specify additionally that its language distinguishes it from the surrounding text. Any element in the TEI scheme may take a **xml:lang** attribute, which specifies both the writing system and the language used by its content (see section *vi.1 Language identification* for discussion of this attribute). Where there is no other applicable element, the element <foreign> may be used to provide a peg onto which the **xml:lang** may be attached.

```
<q>Aren't you confusing <foreign xml:lang="la">post hoc</foreign> with <foreign xml:lang="la">propter
hoc</foreign>?</q> said the Bee Master.
<q>Wax-moth only succeed when weak bees let them in.</q>
```

```
Source: [110]
```

The <foreign> element should not be used to represent foreign words which are mentioned or glossed within the text: for these use the appropriate element from section *3.3.4. Terms, Glosses, Equivalents, and Descriptions* below. Compare the following example sentences:

```
John eats a <foreign xml:lang="fr">croissant</foreign> every morning.
```

```
<mentioned  xml:lang="fr">Croissant</mentioned> is difficult to
pronounce with your mouth full.
```

```
A <term  xml:lang="fr">croissant</term> is a crescent-shaped
piece of light, buttery, pastry that is usually eaten for
breakfast, especially in France.
```

Source: [44]

3.3.2.2 Emphatic Words and Phrases

The <emph> element is provided to mark words or phrases which are *linguistically* emphatic or stressed. Text which is only typographically 'emphasized' falls into the class of highlighted text, and may be tagged with the <hi> element. In printed works, emphasis is generally indicated by devices such as the use of an italic font, a large typeface, or extra wide letter spacing; in manuscripts and typescripts, it is usually indicated by the use of underlining. As the following examples demonstrate, an encoder may choose whether or not to make explicit the particular type of rendition associated with the emphasis by use of the **rend** attribute. If a source text consistently renders a particular feature (e.g. emphasis or words in foreign languages) in a particular way, the rendering associated with that feature may be described in the TEI header using the <rendition> element. The **rend** attribute may then be used to describe examples which deviate from the norm. For example, assuming that the TEI Header has defined a default rendering for the <emph> element, the following encoding would use it:

```
<q>Sex, sir, is <emph>purely</emph> a
question of appetite!</q> Tarr exclaimed.
```

Source: [126]

If on the other hand no such default has been defined for the element, the encoder may specify it informally using the **rend** attribute:

```
<q>What it all comes to is this,</q> he said.

<q>
  <emph  rend="italic">What does Christopher
     Robin do in the morning nowadays?</emph>
</q>
```

Source: [142]

or, if a <rendition> element has been provided in the header (but not necessarily associated with any other element), the **rendition** attribute may be used to point to it:

```
<l>Here Thou, great <name  rend="italics">Anna</name>!
whom three Realms obey,</l>
<l>Doth sometimes Counsel take —
and sometimes <emph  rendition="#italic">Tea</emph>.</l>
<!-- in the header ... -->
<rendition  xml:id="italic"  scheme="css">text-style:italic</rendition>
```

Source: [158]

Further information on the use of the <rendition> element is provided at *2.3.4. The Tagging Declaration*.

The <hi> element is used to mark words or phrases which are highlighted in some way, but for which identification of the intended distinction is difficult, controversial, or impossible. It enables an encoder simply to record the fact of highlighting, possibly describing it by the use of a **rend** or **rendition** attribute, as discussed above, without however taking a position as to the function of the highlighting. This may also be useful if the text is to be processed in two stages: representing simply typographic distinctions during a first pass, and then replacing the <hi> elements with more specific elements in a second pass.

Some simple examples:

```
<hi  rend="gothic">And this Indenture further witnesseth</hi>
that the said <hi  rend="italic">Walter Shandy</hi>, merchant,
in consideration of the said intended marriage ...
```

Source: [187]

In this example, the first highlighted phrase uses black letter or gothic print to mimic the appearance of a legal document, and italic to mark *Walter Shandy* as a name. In a second pass, the elements <head> or <label> might be appropriate for the first use, and the element <name> for the second.

```
The heaviest rain, and snow, and hail, and sleet, could
boast of the advantage over him in only one respect. They
often <hi  rend="quoted">came down</hi> handsomely, and
Scrooge never did.
```

Source: [58]

In this example, the phrase *came down* uses inverted commas to indicate a play on words.[2] In a second pass, the element <soCalled> might be preferred.

3.3.2.3 Other Linguistically Distinct Material

For some kinds of analysis, it may be desirable to encode the linguistic distinctiveness of words and phrases with more delicacy than is allowed by the <foreign> element. The <distinct> element is provided for this purpose. Its attributes allow for additional information characterizing the nature of the linguistic distinction to be made in two distinct ways: the **type** attribute simply assigns a user-defined code of some kind to the word or phrase which assigns it to some register, sub-language, etc. No recommendations as to the set of values for this attribute are provided at this time, as little consensus exists in the field.

Alternatively, the remaining three attributes may be used in combination to place a word or phrase on a three-dimensional scale sometimes used in descriptive linguistics, as for example in Mattheier et al, 1988. The **time** attribute places a word diachronically, for example as archaic, old-fashioned, contemporary, futuristic, etc.; the **space** attribute places a word diatopically, that is, with respect to a geographical classification, for example as national, regional, international, etc.; the **social** attribute places a word diastatically, that is, with respect to a social classification, for example as technical, polite, impolite, restricted, etc. Again, no recommendations are made for the values of these attributes at this time; the encoder should provide a description of the scheme used in the appropriate section of the header (see section *2.3. The Encoding Description*).

Examples:

[2] The Oxford English Dictionary documents the phrase *to come down* in the sense 'to bring or put down; *esp.* to lay down money; to make a disbursement' as being in use, mostly in colloquial or humorous contexts, from at least 1700 to the latter half of the 19th century.

```
Next morning a boy in that dormitory confided to his
bosom friend, a <distinct type="psSlang">fag</distinct> of
Macrea's, that there was trouble in their midst which
King <distinct type="archaic">would fain</distinct> keep
secret.
```

Source: [111]

```
Next morning a boy in that dormitory confided to his
bosom friend, a
<distinct time="1900" space="GB" social="publicschool">fag</distinct>
of Macrea's, that there was trouble in their midst which
King <distinct time="archaic">would fain</distinct> keep
secret.
```

Where more complex (or more rigorous) interpretive analyses of the associations of a word are required, the more detailed and general mechanisms described in chapter *18. Feature Structures* should be preferred to these simple characterizations. It may also be preferable to record the kinds of analysis suggested here by means of the simple annotation element <note> described in section *3.8. Notes, Annotation, and Indexing*, or the element described in section *17.3. Spans and Interpretations*.

3.3.3 Quotation

One form of presentational variation found particularly frequently in written and printed texts is the use of quotation marks. As with the typographic variations discussed in the preceding section, it is generally helpful to separate the encoding of the underlying textual feature (for example, a quotation or a piece of direct speech) from the encoding of its rendering (for example, the use of a particular style of quotation marks).

This section discusses the following elements, all of which are often rendered by the use of quotation marks:

<q> (separated from the surrounding text with quotation marks) contains material which is marked as (ostensibly) being somehow different than the surrounding text, for any one of a variety of reasons including, but not limited to: direct speech or thought, technical terms or jargon, authorial distance, quotations from elsewhere, and passages that are mentioned but not used.

<said> (speech or thought) indicates passages thought or spoken aloud, whether explicitly indicated in the source or not, whether directly or indirectly reported, whether by real people or fictional characters.

@**direct** may be used to indicate whether the quoted matter is regarded as direct or indirect speech.

@**aloud** may be used to indicate whether the quoted matter is regarded as having been vocalized or signed.

<quote> (quotation) contains a phrase or passage attributed by the narrator or author to some agency external to the text.

<cit> (cited quotation) contains a quotation from some other document, together with a bibliographic reference to its source. In a dictionary it may contain an example text with at least one occurrence of the word form, used in the sense being described, or a translation of the headword, or an example.

<mentioned> marks words or phrases mentioned, not used.

<soCalled> contains a word or phrase for which the author or narrator indicates a disclaiming of responsibility, for example by the use of scare quotes or italics.

The elements <mentioned> and <soCalled> are members of the class model.emphLike; the <q> and <said> are members of the class model.qLike in their own right, while <cit> and <quote> are members of model.quoteLike, a subclass of model.qLike. This class is a subclass of model.inter; hence all of these elements are permitted both within and between paragraph-level elements.

The most common and important use of quotation marks is, of course, to mark quotation, by which we mean simply any part of the text attributed by the author or narrator to some agency other than the narrative voice. The <q> element may be used if no further distinction beyond this is judged necessary. If however it is felt necessary to distinguish passages which are in some sense external to the work from passages of direct speech or thought, a more precise element may be chosen from the list above. Typical examples include passages cited from other works, for which the element <quote> may be used, and words or phrases spoken or thought by people or characters within the current work, for which the element <said> may be used. The <soCalled> element is used for cases where the author or narrator distances him or herself from the words in question without however attributing them to any other voice in particular. The <mentioned> element is appropriate for a case where a word or phrase is being discussed in the body of a text rather than forming part of the text directly.

As noted above, if the distinction among these various reasons why a passage is offset from surrounding text cannot be made reliably, or is not of interest, then all quoted matter may simply be marked using the <q> element.

Quotation may be indicated in a printed source by changes in type face, by special punctuation marks (single or double or angled quotes, dashes, etc.) and by layout (indented paragraphs, etc.). If these characteristics are of interest, one or other of the global **rend** or **rendition** attributes discussed in section *1.3.1.1. Global Attributes* may be used to record them.

Quotation marks themselves may, like other punctuation marks, be felt for some purposes to be worth retaining within a text, quite independently of their description by the **rend** attribute. This should generally be done using the appropriate Unicode character, or, if this is not possible, a numeric character reference (see *v.6.1 Character References*).

Alternatively, the encoder may suppress all quotation marks, possibly recording their form using some appropriate set of conventions in the **rend** attribute. Some examples are shown below:

```
<said  rend="PRE lsquo POST rsquo">Who-e debel
you?</said> — he at last said —
<said  rend="PRE lsquo POST rsquo">you no speak-e,
damme, I kill-e.</said> And so saying,
the lighted tomahawk began flourishing
about me in the dark.
```

Source: [140]

```
Adolphe se tourna vers lui :
<said>— Alors, Albert, quoi de neuf?</said>
<said>— Pas grand-chose.</said>
<said>— Il fait beau,</said> dit Robert.
```

```
Adolphe se tourna vers lui :
<said  rend="PRE mdash">Alors,
Albert, quoi de neuf ?</said>
<said  rend="PRE mdash">Pas grand-chose.</said>
<said  rend="PRE mdash">Il fait beau,</said>
dit Robert.
```

Source: [162]

As members of the att.ascribed class, elements <said> and <q> share the following attribute:

att.ascribed provides attributes for elements representing speech or action that can be ascribed to a specific individual.

@**who** indicates the person, or group of people, to whom the element content is ascribed.

This may be used to make explicit who is speaking:

```
Adolphe se tourna vers lui :
<said who="#Adolphe">— Alors, Albert,
quoi de neuf?</said>
<said who="#Albert">— Pas grand-chose.</said>
<said who="#Robert">— Il fait beau,</said>
dit Robert.

<!-- .... -->
<list type="speakers">
  <item xml:id="Adolphe"/>
  <item xml:id="Albert"/>
  <item xml:id="Robert"/>
</list>
```

The **who** attribute may be supplied whether or not an indication of the speaker is given explicitly in the text. It may take the form (as above) of a normalized form of the speaker's name, but its role is to act as a pointer to a location elsewhere in the text where data about each speaker may be supplied. The most appropriate place to place such information is within the participant description component of the TEI Header, as further discussed in *15.2.2. The Participant Description* but for simple cases like the above, a simple list of speakers located in the front or back matter of the text may suffice.

It may also be useful to distinguish representations of speech from representations of thought, in modern printed texts often indicated by a change of typeface. The **aloud** attribute is provided for this purpose, as in this example:

```
<said aloud="true">Oh yes,</said> said Henry,
<said aloud="false">I mean
Gordon Macrae, for example…</said>
<said aloud="false">Jungian
Analyst with Winebox! That's what you called him, you callous bastard,
didn't you? Eh? Eh?</said>
```

Source: [208]

Quoted matter may be embedded within quoted matter, as when one speaker reports the speech of another:

```
<said who="#Wilson">Spaulding, he came down into the office just this day
eight weeks with this very paper in his hand, and he says:—
<said who="#WilsonSpaulding">I wish to the Lord, Mr. Wilson, that I was a
    red-headed man.</said>
</said>
<!-- ... -->
<list type="speakers">
  <item xml:id="Wilson">Wilson</item>
  <item xml:id="WilsonSpaulding">Spaulding reported by Wilson</item>
<!-- ...-->
</list>
```

Source: [62]

Direct speech nested in this way is treated in the same way as elsewhere: a change of rendition may occur, but the same element should be used. An encoder may however choose to distinguish between direct speech which contains quotations from extra-textual matter and direct speech itself, as in the following example:

```
<p>
  <said>The Lord! The Lord! It is Sakya Muni himself,</said> the lama half
sobbed; and under his breath began the wonderful Buddhist
invocation:-<said>
    <quote>
      <l>To Him the Way — the Law — Apart —</l>
      <l>Whom Maya held beneath her heart</l>
      <l>Ananda's Lord — the Bodhisat</l>
    </quote>
     And He is here! The Most Excellent Law is here also. My
     pilgrimage is well begun. And what work! What work!</said>
</p>
```

Source: [112]

Quotations from other works are often accompanied by a reference to their source. The <cit> element may be used to group together the quotation and its associated bibliographic reference, which should be encoded using the elements for bibliographic references discussed in section *3.11. Bibliographic Citations and References*, as in the following example.

```
<div  xml:id="mm01"  type="chapter">
  <head>Chapter 1</head>
  <epigraph>
    <cit>
      <quote>
        <l>Since I can do no good because a woman</l>
        <l>Reach constantly at something that is near it.</l>
      </quote>
      <bibl>
        <title>The Maid's Tragedy</title>
        <author>Beaumont and Fletcher</author>
      </bibl>
    </cit>
  </epigraph>
  <p>Miss Brooke had that kind of beauty which seems to be thrown into
     relief by poor dress...</p>
</div>
```

Source: [68]

Like other bibliographic references, the citation attached to a quotation may be represented simply by a pointer, as in this example:

```
Lexicography has shown little sign of being affected by the
work of followers of J.R. Firth, probably best summarized
in his slogan, <cit>
  <quote>You shall know a word by the company it keeps.</quote>
  <ref>(Firth, 1957)</ref>
</cit>
```

Source: [96]

Unlike most of the other elements discussed in this chapter, direct speech and quotations may frequently contain other high-level elements such as paragraphs or verse lines, as well as being themselves contained by such elements. Three possible solutions exist for this well-known structural problem:

- the quotation is broken into segments, each of which is entirely contained within a paragraph
- the quotation is marked up using stand-off markup
- the quotation boundaries are represented by empty segment boundary delimiter elements

For further discussion and several examples, see chapter *20. Non-hierarchical Structures*.

Finally, in this section, the element <soCalled> is provided for all cases in which quotation marks are used to distance the quoted text from the narrator or speaker. Common examples include the 'scare' quotes often found in newspaper headlines and advertising copy, where the effect is to cast doubts on the veracity of an assertion:

```
<head>PM dodges <soCalled>election threat</soCalled> in interview</head>
```

Source: [192]

The same element should be used to mark a variety of special ironic usages. Some further examples follow:

```
He hated <soCalled>good</soCalled> books.
```

```
<soCalled>Croissants</soCalled> indeed! toast not good enough for you?
```

```
Although Chomsky's decision that all NL
sentences are finite objects was never justified by arguments from
the attested properties of NLs, it did have a certain
<soCalled>social</soCalled> justification. It was commonly assumed in
works on logic until fairly recently that the notion
<mentioned>language</mentioned> is necessarily restricted to finite
strings.
```

Source: [117]

3.3.4 Terms, Glosses, Equivalents, and Descriptions

This section describes a set of textual elements which are used to provide a gloss, alternate identification, or description of something.

Technical terms are often italicized or emboldened upon first mention in printed texts; an explanation or gloss is sometimes given in quotation marks. Linguistic analyses conventionally cite words in languages under discussion in italics, providing a gloss immediately following marked with single quotation marks. Other texts in which individual words or phrases are mentioned (for example, as examples) rather than used may mark them either with italics or with quotation marks, and will gloss them less regularly.

<term> contains a single-word, multi-word, or symbolic designation which is regarded as a technical term.

<gloss> identifies a phrase or word used to provide a gloss or definition for some other word or phrase.

These elements are also members of the class model.emphLike.

A <term> may appear with or without a gloss, as may a <mentioned> element. Where the <gloss> is present, it may be linked to the term it is glossing by means of its **target** attribute. To establish such a link, the encoder should give an **xml:id** value to the <term> or <mentioned> element and provide that id as the value of the **target** attribute on the <gloss> element. The following examples demonstrate this facility: for more discussion of this and other kinds of linkage within TEI documents, see chapter *16. Linking, Segmentation, and Alignment*.

Examples:

```
We may define <term xml:id="TDPv" rend="sc">discoursal point of view</term>
as
<gloss target="#TDPv">the relationship, expressed through discourse
structure, between the implied author or some other addresser,
and the fiction.</gloss>
```

Source: [121]

```
<gloss rend="unmarked" target="#PRSR">A computational device that infers
structure from grammatical strings of words</gloss> is known as a
<term xml:id="PRSR">parser</term>, and much of the history of NLP over the
last 20 years has been occupied with the design of parsers.
```

Source: [81]

```
There is thus a striking accentual difference between a verbal
form like <mentioned xml:id="cw234" xml:lang="grc">eluthemen</mentioned>
<gloss target="#cw234">we were released,</gloss> accented on the
second syllable of the word, and its participial derivative

<mentioned xml:id="cw235" xml:lang="grc">lutheis</mentioned>
<gloss target="#cw235">released,</gloss> accented on the last.
```

Source: [168]

Another group of elements is used to supply different kinds of names for objects described by the TEI. Examples of this are documentation of elements, attributes, classes (and also attribute values where appropriate), and description of glyphs.

<altIdent> (alternate identifier) supplies the recommended XML name for an element, class, attribute, etc. in some language.

<desc> (description) contains a brief description of the intended usage, purpose, or application of its parent element.

<equiv/> (equivalent) specifies a component which is considered equivalent to the parent element, either by co-reference, or by external link.

- @**uri** (uniform resource identifier) references the underlying concept of which the parent is a representation by means of some external identifier
- @**filter** references an external script which contains a method to transform instances of this element to canonical TEI
- @**name** names the underlying concept of which the parent is a representation

Along with the <gloss> element mentioned above, these elements constitute the model.glossLike class.

The <gloss> element may be used to provide a brief explanation for the name of the object if this is not self-explanatory. For example, the specification for the element <ab> used to mark arbitrary blocks of text begins as follows:

```
<elementSpec module="linking" ident="ab">
  <gloss>anonymous block</gloss>
<!-...-->
</elementSpec>
```

A <gloss> may also be supplied for an attribute name or an attribute value in similar circumstances:

```
<valList type="open">
  <valItem ident="susp">
    <gloss>suspension</gloss>
    <desc>the abbreviation provides the first letter(s)
        of the word or phrase, omitting the remainder.</desc>
  </valItem>
  <valItem ident="contr">
    <gloss>contraction</gloss>
    <desc>the abbreviation omits some letter(s) in the middle.</desc>
  </valItem>
<!-...-->
</valList>
```

Note that this is quite distinct from the use of the <desc> element, which contains a full description of the intended semantics for the object.

The <equiv> element is used to document equivalencies between the concept represented by this object and the same concept as described in other schemes or ontologies. The **uri** attribute is used to supply a pointer to some location where such external concepts are defined. For example, to indicate that the TEI <death> element corresponds to the concept defined by the CIDOC CRM category E69, the declaration for the former might begin as follows:

```
<elementSpec module="namesdates" ident="death">
  <equiv name="E69" uri="http://cidoc.ics.forth.gr/"/>
<!-...-->
</elementSpec>
```

The <equiv> element may also be used to map newly-defined elements onto existing constructs in the TEI, using the **filter** and **name** attributes to point to an implementation of the mapping. This is useful when a TEI customization (see *23.2. Personalization and Customization*) defines 'shortcuts' for convenience of data entry or markup readability. For example, suppose that in some TEI customization an element <bo> has been defined which is conceptually equivalent to the standard markup construct **<hi rend='bold'>**. The following declarations would additionally indicate that instances of the <bo> element can be converted to canonical TEI by obtaining a filter from the URI specified, and running the procedure with the name bold. The **mimeType** attribute specifies the language (in this case XSL) in which the filter is written:

```
<elementSpec ident="bo" ns="http://www.example.org/ns/notTEI">
  <equiv
```

```
     filter="http://www.example.com/equiv-filter.xsl"
     mimeType="text/xsl"
     name="bold"/>
  <gloss>bold</gloss>
  <desc>contains a sequence of characters rendered in a bold face.</desc>
<!-- ... -->
</elementSpec>
```

The <altIdent> element is used to provide an alternative name for an object, for example using a different natural language. Thus, the following might be used to indicate that the <abbr> element should be identified using the German word *Abkürzung*:

```
<elementSpec ident="abbr" mode="change">
  <altIdent xml:lang="de">Abkürzung</altIdent>
<!--...-->
</elementSpec>
```

In the same way, the following specification for the <graphic> element indicates that the attribute **url** may also be referred to using the alternate identifier href:

```
<elementSpec ident="graphic" mode="change">
  <attList>
    <attDef mode="change" ident="url">
      <altIdent>href</altIdent>
    </attDef>
<!-- .... -->
  </attList>
</elementSpec>
```

By default, the <altIdent> of a component is identical to the value of its **ident** attribute.

The contents of the <desc> element provide a brief characterization of the intended function of the object being documented in a form that permits its quotation out of context, as in the following example:

```
<elementSpec module="core" ident="foreign">
<!--... -->
  <desc>identifies a word or phrase as belonging to some language other
     than that of the surrounding text. </desc>
<!--... -->
</elementSpec>
```

By convention, a <desc> element begins with a verb such as *contains*, *indicates*, *specifies*, etc. and contains a single clause.

3.3.5 Some Further Examples

As a simple example of the elements discussed here, consider the following sentence:

> On the one hand the *Nibelungenlied* is associated with the new rise of romance of twelfth-century France, the *romans d'antiquité*, the romances of Chrétien de Troyes, and the German adaptations of these works by Heinrich van Veldeke, Hartmann von Aue, and Wolfram von Eschenbach.

A first approximation to the encoding of this sentence might be simply to record the fact that the phrases printed above in italics are highlighted, as follows:

```
On the one hand the <hi  rend="italic">Nibelungenlied</hi> is
associated with the new rise of romance of twelfth-century France,
the <hi  xml:lang="fr"  rend="italic">romans d'antiquité</hi>,
the romances of Chrétien de Troyes, ...
```

This encoding would, however, lose the important distinction between an italicized title and an italicized foreign phrase. Many other phrases might also be italicized in the text, and a retrieval program seeking to identify foreign terms (for example) would not be able to produce reliable results by simply looking for italicized words. Where economic and intellectual constraints permit, therefore, it would be preferable to encode both the function of the highlighted phrases and their appearance, as follows:

```
On the one hand the <title  rend="italic">Nibelungenlied</title>
is associated with the new rise of romance of twelfth-century France,
the <foreign  rend="italic">romans d'antiquité</foreign>, the
romances of Chrétien de Troyes, ...
```

In this example, the decision as to which textual features are distinguished by the highlighting is relatively uncontroversial. As a less straightforward example, consider the use of italic font in the following passage:

> A pretty common case, I believe; in all *vehement* debatings. She says I am *too witty*; Anglicé, *too pert*; I, that she is *too wise*; that is to say, being likewise put into English, *not so young as she has been*: in short, she is grown so much into a *mother*, that she had forgotten she ever was a *daughter*. ...

Clearly, the word *vehement* is not italicized for the same reason as the phrase *not so young as she has been*; the former is emphasized, while the latter is proverbial. It also provides an ironic gloss for the words *too wise*, in the same way as *too pert* glosses *too witty*. The glossed phrases are not, however, technical terms or cited words, but quoted phrases, as if the writer were putting words into her own and her mother's mouths. Finally, the words *mother* and *daughter* are apparently italicized simply to oppose them in the sentence; certainly they do not fit into any of the categories so far proposed as reasons for italicizing. Note also that the word *Anglicé* is not italicized although it is not generally considered an English word.

The following sample encoding for the above passage attempts to take into account all the above points:

```
A pretty common case, I believe; in all <emph>vehement</emph>
debatings. She says I am <q  rend="italic">too witty</q>;
<foreign  xml:lang="la"  rend="roman">Anglicé</foreign>,
<gloss  rend="italic">too pert</gloss>; I, that she is
<q  rend="italic"> too wise</q>; that is to say, being likewise
put into English, <gloss  rend="italic">not so young as she has
been</gloss>: in short, she is grown so much into a
<hi  rend="italic">mother</hi>, that she had forgotten she ever
was a <hi  rend="italic">daughter</hi>.
```

Source: [164]

3.4 Simple Editorial Changes

As in editing a printed text, so in encoding a text in electronic form, it may be necessary to accommodate editorial comment on the text and to render account of any changes made to the text in preparing it. The tags described in this section may be used to record such editorial interventions, whether made by the encoder, by the editor of a printed edition used as a copy text, by earlier editors, or by the copyists of manuscripts.

The tags described here handle most common types of editorial intervention and stereotyped comment; where less structured commentary of other types is to be included, it should be marked using the <note> element described in section *3.8. Notes, Annotation, and Indexing*. Systematic interpretive annotation is also possible using the various methods described in chapter *16. Linking, Segmentation, and Alignment*. The examples given here illustrate only simple cases of editorial intervention; in particular, they permit economical encoding of a simple set of alternative readings of a short span of text. To encode multiple views of large or heterogenous spans of text, the mechanisms described in chapter *16. Linking, Segmentation, and Alignment* should be used. To encode multiple witnesses of a particular text, a similar mechanism designed specifically for critical editions is described in chapter *12. Critical Apparatus*.

For most of the elements discussed here, some encoders may wish to indicate both a responsibility, that is, a code indicating the person or agency responsible for making the editorial intervention in question, and also an indication of the degree of certainty which the encoder wishes to associate with the intervention. Because these requirements are common to many of the elements discussed in this section, they are provided by an attribute class, called att.editLike. All members of this class carry the following optional attributes:

att.editLike provides attributes describing the nature of a encoded scholarly intervention or interpretation of any kind.

- @**cert** (certainty) signifies the degree of certainty associated with the intervention or interpretation.
- @**resp** (responsible party) indicates the agency responsible for the intervention or interpretation, for example an editor or transcriber.
- @**evidence** indicates the nature of the evidence supporting the reliability or accuracy of the intervention or interpretation.
- @**precision** characterizes the precision of the specified intervention or dating.

Many of the elements discussed here can be used in two ways. Their primary purpose is to indicate that the text encoded as the element's content represents an editorial intervention (or non-intervention) of a specific kind, indicated by the element itself. However, pairs or other meaningful groupings of such elements can also be supplied, wrapped within a special purpose <choice> element:

<choice> groups a number of alternative encodings for the same point in a text.

This element enables the encoder to represent for example a text in its 'original' uncorrected and unaltered form, alongside the same text in one or more 'edited' forms. This usage permits software to switch automatically between one 'view' of a text and another, so that (for example) a stylesheet may be set to display either the text in its original form or after the application of editorial interventions of particular kinds.

Elements which can be combined in this way constitute the model.choicePart class. The default members of this class are <sic>, <corr>, <reg>, <orig>, <unclear>, <add>, and <del>; their functions and usage are described further below.

Three categories of editorial intervention are discussed in this section:

- indication or correction of apparent errors
- indication or regularization of variant, irregular, non-standard, or eccentric forms
- editorial additions, suppressions, and omissions

A more extended treatment of the use of these tags in transcriptional and editorial work is given in chapter *11. Representation of Primary Sources*.

3.4.1 Apparent Errors

When the copy text is manifestly faulty, an encoder or transcriber may elect simply to correct it without comment, although for scholarly purposes it will often be more generally useful to record both the correction and the original state of the text. The elements described here enable all three approaches, and allows the last to be done is such a way as make it easy for software to present either the original or the correction.

<sic> (latin for thus or so) contains text reproduced although apparently incorrect or inaccurate.

<corr> (correction) contains the correct form of a passage apparently erroneous in the copy text.

The following examples show alternative treatment of the same material. The copy text reads:

> Another property of computer-assisted historical research is that data modelling must permit any one textual feature or part of a textual feature to be a part of more than one information model and to allow the researcher to draw on several such models simultaneously, for example, to select from a machine-readable text those marginal comments which indicate that the date's mentioned in the main body of the text are incorrect.

An encoder may choose to correct the typographic error, either silently or with an indication that a correction has been made, as follows:

```
… marginal comments which indicate that the <corr>dates</corr>
mentioned in the main body of the text are incorrect.
```

Alternatively, the encoder may simply record the typographic error without correcting it, either without comment or with a <sic> element to indicate the error is not a transcription error in the encoding:

```
… marginal comments which indicate that the <sic>date's</sic>
mentioned in the main body of the text are incorrect.
```

If the encoder elects both to record the original source text and to provide a correction for the sake of word-search and other programs, both <sic> and <corr> are used, wrapped in a <choice>:

```
… marginal comments which indicate that the
<choice>
  <corr>dates</corr>
  <sic>date's</sic>
</choice> mentioned in the main body of the text are
incorrect.
```

The <sic> and <corr> elements can appear in either order.

If it is desired to indicate the person or edition responsible for the emendation, this might be done as follows:

```
… marginal comments which indicate that the
<choice>
  <corr  resp="#msm">dates</corr>
  <sic>date's</sic>
```

```
</choice> mentioned in the main body of the text are
incorrect.

<!-- within the header for this document ... -->
<respStmt>
  <resp>editor</resp>
  <name  xml:id="msm">C.M. Sperberg McQueen</name>
</respStmt>
```

Here the **resp** attribute has been used to indicate responsibility for the correction. Its value (**#msm**) is an example of the pointer values discussed in section *3.6. Simple Links and Cross-References*; in this case, it points to a <name> element within the TEI Header, but any element might be indicated in this way, including for example a <person> element (if the module described in *13. Names, Dates, People, and Places* has been included), or one of the bibliographic elements described in *3.11. Bibliographic Citations and References*, if the correction has been taken from some other source. The **resp** attribute is available for all elements which are part of the att.editLike class. The same class makes available a **cert** attribute,which may be used to indicate the degree of editorial confidence in a particular correction, as in the following example:

```
An <choice>
  <corr  cert="high">Autumn</corr>
  <sic>Antony</sic>
</choice> it was,
That grew the more by reaping
```

Source: [174]

See further the discussion in section *11.3.3. Correction and Conjecture*.

Where, as here, the correction takes the form of adding text not otherwise present in the text being encoded, the encoder should use the <corr> element. Where the correction is present in the text being encoded, and consists of some combination of visible additions and deletions, the elements <add> or <del> should be used: see further section *3.4.3. Additions, Deletions, and Omissions* below. Where the correction takes the form of addition of material not present in the original because of physical damage or illegibility, the <supplied> element may be used. Where the 'correction' is simply a matter of expanding an abbreviation the <ex> element may be used. These and other elements to support the detailed encoding of authorial or scribal interventions of this kind are all provided by the module described in chapter *11. Representation of Primary Sources*.

3.4.2 Regularization and Normalization

When the source text makes extensive use of variant forms or non-standard spellings, it may be desirable for a number of reasons to regularize it: that is, to provide 'standard' or 'regularized' forms equivalent to the non-standard forms.[3]

As with other such changes to the copy text, the changes may be made silently (in which case the TEI header should specify the types of silent changes made) or may be explicitly marked using the following elements:

<reg> (regularization) contains a reading which has been regularized or normalized in some sense.

<orig> (original form) contains a reading which is marked as following the original, rather than being normalized or corrected.

<choice> groups a number of alternative encodings for the same point in a text.

[3] In some contexts, the term *regularization* has a narrower and more specific significance than that proposed here: the <reg> element may be used for any kind of regularization, including normalization, standardization, and modernization.

Typical applications for these elements include the production of editions intended for student or lay readers, linguistic research in which spelling or usage variation is not the main question at issue, production of spelling dictionaries, etc.

Consider this 16th-century text:

> how godly a dede it is to overthrowe so wicked a race the world may judge: for my part I thinke there canot be a greater sacryfice to God.

An encoder may choose to preserve the original spelling of this text, but simply flag it as nonstandard by using the <orig> element with no attributes specified, as follows:

```
<p>...how godly a <orig>dede</orig> it is to
<orig>overthrowe</orig> so wicked a race the
world may judge: for my part I <orig>thinke</orig>
there <orig>canot</orig> be a greater
<orig>sacryfice</orig> to God</p>
```

Alternatively, the encoder may simply indicate that certain words have been modernized by using the <reg> element with no attributes specified, as follows:

```
<p>...how godly a
<reg>deed</reg> it is to <reg>overthrow</reg> so wicked a race the
world may judge: for my part I <reg>think</reg>
there <reg>cannot</reg> be a greater
<reg>sacrifice</reg> to God.</p>
```

Alternatively, the encoder may elect to record both old and new spellings, so that (for example) the same electronic text may serve as the basis of an old- or new-spelling edition:

```
<p>...how godly a <choice>
   <orig>dede</orig>
   <reg>deed</reg>
  </choice> it is to
<choice>
   <orig>overthrowe</orig>
   <reg>overthrow</reg>
  </choice> so wicked a race the
world may judge: for my part I <choice>
   <orig>thinke</orig>
   <reg>think</reg>
  </choice>
there <choice>
   <orig>canot</orig>
   <reg>cannot</reg>
  </choice> be a greater
<choice>
   <orig>sacryfice</orig>
   <reg>sacrifice</reg>
  </choice> to God.</p>
```

Source: [29]

As elsewhere, the **resp** attribute may be used to specify the agency responsible for the regularization.

3.4.3 Additions, Deletions, and Omissions

The following elements are used to indicate when words or phrases have been omitted from, added to, or marked for deletion from, a text. Like the other editorial elements, they allow for a wide range of editorial practices:

<gap> indicates a point where material has been omitted in a transcription, whether for editorial reasons described in the TEI header, as part of sampling practice, or because the material is illegible, invisible, or inaudible.

@**reason** gives the reason for omission. Sample values include **sampling**, **illegible**, **inaudible**, **irrelevant**, **cancelled**, **illegible**.

<unclear> contains a word, phrase, or passage which cannot be transcribed with certainty because it is illegible or inaudible in the source.

@**reason** indicates why the material is hard to transcribe.

<add> (addition) contains letters, words, or phrases inserted in the text by an author, scribe, annotator, or corrector.

<del> (deletion) contains a letter, word, or passage deleted, marked as deleted, or otherwise indicated as superfluous or spurious in the copy text by an author, scribe, annotator, or corrector.

Encoders may choose to omit parts of the copy text for reasons ranging from illegibility of the source or impossibility of transcribing it, to editorial policy, e.g. a systematic exclusion of poetry or prose from an encoding. The full details of the policy decisions concerned should be documented in the TEI Header (see section *2.3. The Encoding Description*). Each place in the text at which omission has taken place should be marked with a <gap> element, with optionally further information about the reason for the omission, its extent, and the person or agency responsible for it, as in the following examples:

```
<gap reason="illegible" unit="words" extent="2"/>
```

```
<gap reason="overwriting illegible" extent="8" unit="chars"/>
```

The <desc> element may be used to supply a description of the material omitted, where that is considered useful:

```
<gap reason="sampling" extent="120" unit="lines">
  <desc>irrelevant commentary</desc>
</gap>
```

```
… Their arrangement with respect to Jupiter and to each other was as follows:
<gap reason="sampling" extent="2" unit="cm">
  <desc>astrological figure</desc>
</gap>
That is, there were two stars on the easterly side and one to the west; …
```

Source: [77]

The <add> and <del> elements may be used to record where words or phrases have been added or deleted in the copy text. They are not appropriate where longer passages have been added or deleted, which span

several elements; for these, the elements <addSpan> and <delSpan> described in chapter *11.3.4. Additions and Deletions* must be used.

Additions to a text may be recorded for a number of reasons. Sometimes they are marked in a distinctive way in the source text, for example by brackets or insertion above the line (supralinear insertion), as in the following example, taken from a 19th century manuscript:

```
The story I am going to relate is true as to its main facts,
and as to the consequences <add  place="supralinear">of
these facts</add> from which this tale takes its title.
```

Source: [78]

The <add> element should not be used to mark editorial changes, such as supplying a word omitted by mistake from the source text or a passage present in another version. In these cases, either the <corr> or <supplied> tags should be used, as discussed above in section *3.4.1. Apparent Errors*, and in section *11.3.3. Correction and Conjecture*, respectively.

The <unclear> element is used to mark passages in the original which cannot be read with confidence, or about which the transcriber is uncertain for other reasons, as for example when transcribing a partially inaudible or illegible source. Its **reason** and **resp** attributes are used, as with the <gap> element, to indicate the cause of uncertainty and the person responsible for the conjectured reading.

For example:

```
<l>And where the sandy mountain Fenwick scald</l>
<l>
  <unclear  reason="ink blot">The</unclear> sea between
yet hence his pray'r prevail'd
</l>
```

Source: [139]

or from a spoken text:

```
<p>... and then <unclear  reason="passing truck">marbled queen</unclear>...</p>
```

Where the material affected is entirely illegible or inaudible, the <gap> element discussed above should be used in preference.

The <del> element is used to mark material which is deleted in the source but which can still be read with some degree of confidence, as opposed to material which has been omitted by the encoder or transcriber either because it is entirely illegible or for some other reason. This is of particular importance in transcribing manuscript material, though deletion is also found in printed texts, sometimes for humorous purposes:

```
<l>One day I will sojourn to your shores</l>
<l>I live in the middle of England</l>
<l>But!</l>
<l>Norway! My soul resides in your watery
<del  rend="overstrike">fiords fyords fiiords</del>
</l>
<l>Inlets.</l>
```

Source: [196]

The **rend** attribute may be used to distinguish different methods of deletion in manuscript or typescript material, as in this line from the typescript of Eliot's *Waste Land*:

```
<l>
  <del  rend="overtyped">Mein</del> Frisch
<del  type="overstrike">schwebt</del> weht der Wind
</l>
```

Deletion in manuscript or typescript is often associated with addition:

```
<l>
  <del  rend="overstrike">Inviolable</del>
  <add  place="infralinear">Inexplicable</add>
splendour of Corinthian white and gold
</l>
```

Source: [70]

The <subst> element discussed in *11.3.5. Substitutions* provides a way of grouping additions and deletions of this kind.

The <del> element should not be used where the deletion is such that material cannot be read with confidence, or read at all, or where the material has been omitted by the transcriber or editor for some other reason. Where the material deleted cannot be read with confidence, the <unclear> tag should be used with the **reason** attribute indicating that the difficulty of transcription is due to deletion. Where material has been omitted by the transcriber or editor, this may be indicated by use of the <gap> element. A deletion in which some parts may be read but not others may thus be represented by one or more <gap> elements intermingled with text, all contained by a <del> element.

3.5 Names, Numbers, Dates, Abbreviations, and Addresses

This section describes a number of textual features which it is often convenient to distinguish from their surrounding text. Names, dates, and numbers are likely to be of particular importance to the scholar treating a text as source for a database; distinguishing such items from the surrounding text is however equally important to the scholar primarily interested in lexis.

The treatment of these textual features proposed here is not intended to be exhaustive: fuller treatments for names, numbers, measures, and dates are provided in the names and dates module (see chapter *13. Names, Dates, People, and Places*).

3.5.1 Referring Strings

A referring string is a phrase which refers to some person, place, object, etc. Two elements are provided to mark such strings:

<rs> (referencing string) contains a general purpose name or referring string.

<name> (name, proper noun) contains a proper noun or noun phrase.

Where it is thought useful to do so, the kind of object referred to may be specified using the **type** attribute. Examples include:

```
<p>
  <q>My dear
  <rs  type="person">Mr. Bennet</rs>
  </q>, said his lady to
him one day, <q>have you heard that <rs  type="place">Netherfield Park</rs> is let at last?</q>
</p>
```

Source: [7]

```
<p>Collectors of water-rents were appointed by the
<rs  type="organization">Watering Committee</rs>.
They were paid a commission not exceeding four per
cent, and gave bond.</p>
```

Source: [3]

```
<p>It being one of the principles of the
<rs  type="org">Circumlocution Office</rs> never, on any
account whatsoever, to give a straightforward answer,
<rs  type="person">Mr Barnacle</rs> said, <q>Possibly.</q>
</p>
```

Source: [59]

As the following example shows, the <rs> element may be used for any reference to a person, place, etc., not only to references in the form of a proper noun or noun phrase.

```
<p>
  <q>My dear <rs  type="person">Mr. Bennet</rs>
  </q>, said
<rs  type="person">his lady</rs> to him one day ...
</p>
```

The <name> element by contrast is provided for the special case of referencing strings which consist only of proper nouns; it may be used synonymously with the <rs> element, or nested within it if a referring string contains a mixture of common and proper nouns. The following example shows an alternative way of encoding the short sentence from *Pride and Prejudice* quoted above:

```
<p>
  <q>My dear <name  type="person">Mr. Bennet</name>,</q> said <rs  type="person">his lady</rs> to him one day,
<q>have you heard that <name  type="place">Netherfield Park</name> is let at last?</q>
</p>
```

As the following example shows, a proper name may be nested within a referring string:

```
<rs>His Excellency the Life President, <name>Ngwazi Dr H. Kamuzu Banda</name>
</rs>
```

Source: [136]

Simply tagging something as a name is generally not enough to enable automatic processing of personal names into the canonical forms usually required for reference purposes. The name as it appears in the text may be inconsistently spelled, partial, or vague. Moreover, name prefixes such as *van* or *de la* may or may not be included as part of the reference form of a name, depending on the language and country of origin of the bearer.

Two issues arise in this context: firstly, there may be a need to encode a regularised form of a name, distinct from the actual form in the source to hand; secondly, there may be a need to identify the particular person, place, etc. referred to by the name, irrespective of whether the name itself is normalized or not. The element

<reg>, introduced in *3.4.2. Regularization and Normalization* is provided for the former purpose; the attribute **key** for the latter.

The **key** attribute is common to all members of the att.naming class and is defined as follows:

att.naming provides attributes common to elements which refer to named persons, places, organizations etc.

@**key** provides an external means of locating a full definition for the entity (or entities) being named, such as a database record key or other token.

Its main use is as a means of gathering together all references to the same individual or location scattered throughout a document:

```
<p>
  <q>My dear <rs  key="BENM1"  type="person"> Mr. Bennet</rs>,</q>
said <rs  key="BENM2"  type="person">his lady</rs> to him one day,
<q>have you heard that <rs  key="NETP1"  type="place">Netherfield
       Park</rs> is let at last?</q>
</p>
```

```
<p>
  <name  key="VOM1"  type="person">Mme. de Volanges</name> marie <rs  key="VOM2">sa fille</rs>: c'est encore un
secret;
mais elle m'en a fait part hier.
</p>
```

Source: [115]

The value of the **key** attribute may be an unexpanded code, as in the examples above, with no significance. More usually however, it will reference some other resource providing more information about the entity named by the element, which might itself include a number of alternative forms of name for the entity in question.

This use should be distinguished from the use of a nested <reg> (regularization) element to provide the standard form of a referring string, as in this example:

```
<p>My personal life during
the administration of <rs  key="POJA1"  type="person">Col. Polk
    (<reg>Polk, James K.</reg>)</rs> has but poorly compensated me for the
suspended enjoyments and pursuits of private and professional
spheres</p>
```

Source: [51]

The <choice> element discussed in *3.4. Simple Editorial Changes* may be used if it is desired to record both a normalized form of a name and the name used in the source being encoded:

```
<p>
  <name  key="WADLM1"  type="person">
    <choice>
      <orig>Walter de la Mare</orig>
      <reg>de la Mare, Walter</reg>
    </choice>
  </name>
```

```
was born at <name  key="Ch1"  type="place">Charlton</name>, in
<name  key="KT1"  type="county">Kent</name>, in 1873.
</p>
```

Source: [190]

The <index> element discussed in *3.8.2. Index Entries* may be more appropriate if the function of the regularization is to provide a consistent index:

```
<p>
  <name  type="place">Montaillou</name> is not a large parish.
At the time of the events which led to
<name  type="person">Fournier<index>
     <term>Benedict XII, Pope of Avignon (Jacques Fournier)</term>
   </index>
  </name>'s
investigations, the local population consisted of between 200 and 250 inhabitants.
</p>
```

Source: [116]

Although adequate for many simple applications, these methods have two inconveniences: if the name occurs many times, then its regularised form must be repeated many times; and the burden of additional XML markup in the body of the text may be inconvenient to maintain and complex to process. For applications such as onomastics, relating to persons or places named rather than the name itself, or wherever a detailed analysis of the component parts of a name is needed, the specialized elements described in chapter *13. Names, Dates, People, and Places* or the analytical tools described in chapter *18. Feature Structures* should be used.

3.5.2 Addresses

These Guidelines propose the following elements to distinguish postal and electronic addresses:

<address> contains a postal address, for example of a publisher, an organization, or an individual.

<email> (electronic mail address) contains an e-mail address identifying a location to which e-mail messages can be delivered.

These two elements constitute the class of model.addressLike elements; for other kinds of address this class may be extended by adding new elements if necessary.

These Guidelines provide no particular means for encoding the substructure of an email address (for example, distinguishing the local part from the domain part), nor of distinguishing personal email addresses from generic or fictitious ones.

```
<email>editors@tei-c.org</email>
```

The simplest way of encoding a postal address is to regard it as a series of distinct lines, just as they might be written on an envelope. The following element supports this view:

<addrLine> (address line) contains one line of a postal address.

Here is an example of a postal address encoded using this approach:

```
<address>
  <addrLine>110 Southmoor Road,</addrLine>
  <addrLine>Oxford OX2 6RB,</addrLine>
  <addrLine>UK</addrLine>
</address>
```

Alternatively, an address may be encoded as a structure of more semantically rich elements. The class model.addrPart element class identifies a number of such possible components:

<street> a full street address including any name or number identifying a building as well as the name of the street or route on which it is located.

<name> (name, proper noun) contains a proper noun or noun phrase.

@**type** indicates the type of the object which is being named by the phrase.

<postCode> (postal code) contains a numerical or alphanumeric code used as part of a postal address to simplify sorting or delivery of mail.

<postBox> (postal box or post office box) contains a number or other identifier for some postal delivery point other than a street address.

model.nameLike groups elements which name or refer to a person, place, or organization.

model.persNamePart groups elements which form part of a personal name.

model.placeNamePart groups elements which form part of a place name.

Any number of elements from the model.addrPart class may appear within an address and in any order. None of them is required.

Where code letters are commonly used in addresses (for example, to identify regions or countries) a useful practice is to supply the full name of the region or country as the content of the element, but to supply the abbreviatory code as the value of the global **n** attribute, so that (for example) an application preparing formatted labels can readily find the required information. Other components of addresses may be represented using the general-purpose <name> element or (when the additional module for names and dates is included) the more specialized elements provided for that purpose.

Using just the elements defined by the core module, the above address could thus be represented as follows:

```
<address>
  <street>110 Southmoor Road</street>
  <name  type="city">Oxford</name>
  <postCode>OX2 6RB</postCode>
  <name  type="country">United Kingdom</name>
</address>
```

The order of elements within an address is highly culture-specific, and is therefore unconstrained:

```
<address>
  <name  type="org">Università di Bologna</name>
  <name  type="country">Italy</name>
  <postCode>40126</postCode>
  <name  type="city">Bologna</name>
  <street>via Marsala 24</street>
</address>
```

For further discussion of ways of regularizing the names of places, see section *3.5. Names, Numbers, Dates, Abbreviations, and Addresses.* A full postal address may also include the name of the addressee, tagged as above using the general purpose <name> element.

When a schema includes the names and dates module discussed in chapter *13. Names, Dates, People, and Places*, a large number of more specific elements such as <country> or <settlement> will be available from the class model.addrPart. The above example might then be encoded as follows:

```
<address>
  <street>110 Southmoor Road</street>
  <settlement>Oxford</settlement>
  <postCode>OX2 6RB</postCode>
  <country>United Kingdom</country>
</address>
```

3.5.3 Numbers and Measures

This section describes elements provided for the simple encoding of numbers and measurements and gives some indication of circumstances in which this may usefully be done. The following phrase level elements are provided for this purpose:

<num> (number) contains a number, written in any form.

@**type** indicates the type of numeric value.

@**value** supplies the value of the number in standard form.

<measure> contains a word or phrase referring to some quantity of an object or commodity, usually comprising a number, a unit, and a commodity name.

@**type** specifies the type of measurement in any convenient typology.

<measureGrp> (measure group) contains a group of dimensional specifications which relate to the same object, for example the height and width of a manuscript page.

Like names or abbreviations, numbers can occur virtually anywhere in a text. Numbers are special in that they can be written with either letters or digits (*twenty-one*, *xxi*, and *21*) and their presentation is language-dependent (e.g. English *5th* becomes Greek *5.*; English *123,456.78* equals French *123.456,78*).

For many kinds of application, e.g. natural-language processing or machine translation, numbers are not regarded as 'lexical' in the same way as other parts of a text. For these and other applications, the <num> element provides a convenient method of distinguishing numbers from the surrounding text. For other kinds of application, numbers are only useful if normalized: here the <num> element is useful precisely because it provides a standardized way of representing a numerical value.

For example:

```
<num value="33">xxxiii</num>
<num type="cardinal" value="21">twenty-one</num>
<num type="percentage" value="10">ten percent</num>
<num type="percentage" value="10">10%</num>
<num type="ordinal" value="5">5th</num>
```

```
<num type="fraction" value="0.5">one half</num>
<num type="fraction" value="0.5">1/2</num>
```

In its fullest form, a measure consists of a number, a phrase expressing units of measure and a phrase expressing the commodity being measured, though not all of these components need be present in every case. It may be helpful to distinguish measures from surrounding text for two reasons. Firstly, a measure may be expressed using a particular notation or system of abbreviations which the encoder does not wish to regard as

lexical. Secondly, a quantitative application may wish to distinguish and normalize the internal components of a measure, in order to perform calculations on them.

Consider, as an example of the first case, the following list of Celia's charms, in which the encoder has chosen to make explicit the measurements:

```
<div n="2">
  <list type="gloss">
    <label>Age</label>
    <item>Unimportant</item>
    <label>Head</label>
    <item>Small and round</item>
    <label>Eyes</label>
    <item>Green</item>
    <label>Complexion</label>
    <item>White</item>
    <label>Hair</label>
    <item>yellow</item>
    <label>Features</label>
    <item>Mobile</item>
    <label>Neck</label>
    <item>
      <measure>13¾"</measure>
    </item>
    <label>Upper arm</label>
    <item>
      <measure>11"</measure>
    </item>
<!--...-->
  </list>
<!-- ... -->
</div>
```

Source: [11]

In the same way, it may be convenient to mark representations of currency which might otherwise be misinterpreted as lexical:

```
<p>...the sum of
<measure type="currency">12s 6d</measure>...</p>
```

In general, normalization of a measure will require specification of one or more of its three parts: the quantity, the units, and possibly also the commodity being measured. This is accomplished by supplying values for the three attributes **quantity**, **unit**, and **commodity**, which are supplied by the att.measurement class:

att.measurement provides attributes to represent a regularized or normalized measurement.

- @**quantity** specifies the number of the specified units that comprise the measurement
- @**unit** indicates the units used for the measurement, usually using the standard symbol for the desired units.
- @**commodity** indicates the substance that is being measured

With these attributes, the measurement of Celia's neck may be specified in a normalized form:

```
<measure quantity="13.75" unit="in">13¾"</measure>
```

Such techniques are particularly useful when representing historical data such as inventories:

```
<list>
  <item>
    <measure
       type="volume"
       quantity="2"
       unit="bag"
       commodity="hops"> ii bags hops </measure>
  </item>
  <item>
    <measure
       type="volume"
       quantity="6"
       unit="truss"
       commodity="cloth"> six trusses Woolen and linen goods </measure>
  </item>
  <item>
    <measure
       type="weight"
       quantity="5"
       unit="ton"
       commodity="coal"> 5 tonnes coale
    </measure>
  </item>
</list>
```

Source: [205]

The <measureGrp> element is provided as a means of grouping several related measurements together, either because the measurement involves several dimensions (for example height and width) or to avoid the need to repeat all the normalizing attributes:

```
<measureGrp  type="volume"  unit="in">
  <measure  type="height"  quantity="14">xiv</measure>
  <measure  type="width"  quantity="5">v</measure>
  <measure  type="depth"  quantity="10">x</measure>
</measureGrp>
```

3.5.4 Dates and Times

Dates and times, like numbers, can appear in widely varying culture- and language-dependent forms, and can pose similar problems in automatic language processing. Such elements constitute the model.dateLike class, of which the default members are:

<date> contains a date in any format.

@**calendar** indicates the system or calendar to which the date belongs.

<time> contains a phrase defining a time of day in any format.

These elements have some additional attributes by virtue of being members of the att.datable and att.duration classes which, in turn, are members of the att.datable.w3c and att.duration.w3c classes. In particular, the **when** attribute will be discussed here:

att.datable.w3c provides attributes for normalization of elements that contain datable events using the W3C datatypes.

@**when** supplies the value of a date or time in a standard form, e.g. yyyy-mm-dd.

Dates can occur virtually anywhere in a text, but in some contexts (e.g. bibliographic citations) their encoding is recommended or required rather than optional. Times can also appear anywhere but are generally optional.

Partial dates or times (e.g. *1990*, *September 1990*, *twelvish*) can be expressed in the **when** attribute by simply omitting a part of the value supplied. Imprecise dates or times (for example *early August*, *some time after ten and before twelve*) may be expressed as date or time ranges.

Where the certainty (i.e. reliability) of the date or time itself is in question, rather than its precision, the encoder should record this fact using the mechanisms discussed in chapter *21. Certainty and Responsibility*.

These mechanisms are useful primarily for fully specified dates or times known with certainty. If component parts of dates or times are to be marked up, or if a more complex analysis of the meaning of a temporal expression is required, the techniques described in chapter *13. Names, Dates, People, and Places* should be used in preference to the simple method outlined here.

The **when** attribute is a useful way of normalizing or disambiguating dates and times which can appear in many formats, as the following examples show:

```
<date when="1980-02-12">12/2/1980</date>
```

```
Given on the <date when="1977-06-12">Twelfth Day of June
in the Year of Our Lord One Thousand Nine Hundred and
Seventy-seven of the Republic the Two Hundredth and first
and of the University the Eighty-Sixth.</date>
```

The **when** attribute always supplies a normalized representation of the date given as content of the <date> element. The format used should be a valid W3C schema datatype.[4] Some typical examples follow:

```
<date when="2001">The
year 2001</date>
<date when="2001-09">September 2001</date>
<date when="2001-09-11">11 Sept 01</date>
<date when="--09-11">9/11</date>
<date when="--09">September</date>
<date when="---11">Eleventh of the month</date>
<time when="08:48:00">8:48</time>
<date when="2001-09-11T12:48:00">Sept 11th, 12 minutes before 9 am</date>
```

[4] The datatypes are taken from the W3C Recommendation *XML Schema Part 2: Datatypes Second Edition.* The permitted datatypes are:

- date
- gYear
- gMonth
- gDay
- gYearMonth
- gMonthDay
- time
- dateTime

There is one exception: these Guidelines permit a time to be expressed as only a number of hours, or as a number of hours and minutes, as per ISO 8601:2004 section 4.2.2.3 and 4.3.3. The W3C time and dateTime datatypes require that the minutes and seconds be included in the normalized value if they are to be correctly processed for example when sorting.

Note in the last example the use of a normalized representation for the date string which includes a time: this example could thus equally well be tagged using the <time> element.

The following examples demonstrate the use of the <date> element to mark a period of time:

```
<p>Those five years —
<date  from="1918"  to="1923">1918 to 1923</date>
— had been, he suspected,
somehow very important.</p>
```

Source: [209]

```
<p>The Eddic poems are preserved in a unique
manuscript (Codex Regius 2365) from <date  notBefore="1250"  notAfter="1300">the second half of the thirteenth
    century</date>, and <title>Hervarar
    saga</title> dates from <date  when="1300">around 1300</date>.</p>
```

The **calendar** attribute may be used to specify a date in any calendar system; if the **when** attribute is also supplied, it should specify the equivalent date in the Gregorian calendar.

3.5.5 Abbreviations and Their Expansions

It is sometimes desirable to mark abbreviations in the copy text, whether to trigger special processing for them, to provide the full form of the word or phrase abbreviated, or to allow for different possible expansions of the abbreviation. Abbreviations may be transcribed as they stand, or expanded; they may be left unmarked, or marked using these tags:

<abbr> (abbreviation) contains an abbreviation of any sort.

<expan> (expansion) contains the expansion of an abbreviation.

The <abbr> element is useful as a means of distinguishing semi-lexical items such as acronyms or jargon:

```
We can sum up the above discussion as follows: the identity of a
<abbr>CC</abbr> is defined by that calibration of values which
motivates the elements of its <abbr>GSP</abbr>; ...
```

Source: [95]

```
Every manufacturer of <abbr>3GL</abbr> or <abbr>4GL</abbr>
languages is currently nailing on <abbr>OOP</abbr> extensions.
```

Source: [67]

The **type** attribute may be used to distinguish types of abbreviation by their function:

```
<abbr  type="title">Dr.</abbr>
<abbr  type="initial">M.</abbr> Deegan is
the Director of the <abbr  type="acronym">CTI</abbr> Centre for Textual Studies.
```

Abbreviations such as *Dr. M.* above may be treated as two abbreviations, as above, or as one:

```
<abbr>Dr. M.</abbr> Deegan is
the Director of the <abbr>CTI</abbr> Centre for Textual Studies.
```

The <expan> element may be used simply to record that an abbreviation has been silently expanded by the encoder, perhaps for reasons of house style or editorial policy. It should always include the whole of an abbreviated phrase or word. More usually however this will be combined with the <abbr> element inside a <choice> element to record both the abbreviation and its expansion:

```
 the
<choice>
  <expan>World Wide Web Consortium</expan>
  <abbr>W3C</abbr>
</choice>
```

Nested abbreviations may also be handled in this way:

```
<choice>
  <abbr>RELAXNG</abbr>
  <expan>regular
     language for <choice>
      <abbr>XML</abbr>
      <expan>extensible markup
            language</expan>
    </choice>, next
     generation</expan>
</choice>
```

Abbreviation is a particularly important feature of manuscript and other source materials, the transcription of which needs more detailed treatment than is possible using these simple elements. A more detailed set of recommendations is discussed in *11.3. Altered, Corrected, and Erroneous Texts*, which includes additional elements made available for the purpose by the transcr module.

3.6 Simple Links and Cross-References

Cross-references or links between one location in a document and one or more other locations, either in the same or different XML documents, may be encoded using the elements <ptr> and <ref>, as discussed in this section. These elements both 'point' from one location in a document, the place that the element itself appears, to another (or to several), specified by the **target** attribute. Linkages of several other kinds are also provided for in these guidelines; see further chapter *16. Linking, Segmentation, and Alignment*.

The value of the **target** attribute, wherever it appears, provides a way of pointing to some other element using a method standardized by the W3C consortium, and known as the XPointer mechanism. This permits a range of complexity, from the very simple (a reference to the value of the target element's **xml:id** attribute) to the more complex usage of a full URI with embedded XPointers. For example, the source of the following paragraph looks something like this:

```
<p>For an introduction
to the use of links in general, see <ptr  target="#SA"/>; for the
complete XPointer specification, see <ptr
     target="http://www.w3.org/TR/xptr-framework/"/>,
<ptr  target="http://www.w3.org/TR/xptr-element/"/>,
```

```
<ptr  target="http://www.w3.org/TR/xptr-xmlns/"/>, and
<ptr
     target="http://www.w3.org/TR/xptr-xpointer/#xpointer(id('chum')/quote)"/>;
for a discussion of TEI schemes for XPointer, see
<ptr  target="#SATS"/>.</p>
```

Alternatively, if no explicit link is to be encoded, but it is simply required to mark the phrase as a cross-reference, the <ref> element may be used without a **target** attribute.

For an introduction to the use of links in general, see *16. Linking, Segmentation, and Alignment*; for the complete XPointer specification, see `http://www.w3.org/TR/xptr-framework/`, `http://www.w3.org/TR/xptr-element/`, `http://www.w3.org/TR/xptr-xmlns/`, and `http://www.w3.org/TR/xptr-xpointer/#xpointer(id('chum')/quote)`; for a discussion of TEI schemes for XPointer, see *16.2.4. TEI XPointer Schemes*.

<ptr/> (pointer) defines a pointer to another location.
- @**target** specifies the destination of the pointer by supplying one or more URI References
- @**cRef** (canonical reference) specifies the destination of the pointer by supplying a canonical reference from a scheme defined in a <refsDecl> element in the TEI header

<ref> (reference) defines a reference to another location, possibly modified by additional text or comment.
- @**target** specifies the destination of the reference by supplying one or more URI References
- @**cRef** (canonical reference) specifies the destination of the reference by supplying a canonical reference from a scheme defined in a <refsDecl> element in the TEI header

The elements <ptr> and <ref> are the default members of the phrase-level model class model.ptrLike. As members of the class att.pointing, they also carry the following attributes:

att.pointing defines a set of attributes used by all elements which point to other elements by means of one or more URI references.
- @**type** categorizes the pointer in some respect, using any convenient set of categories.
- @**evaluate** specifies the intended meaning when the target of a pointer is itself a pointer.

The two elements may be used in the same way; the difference between them is simply that while the <ptr> element is empty, the <ref> element may contain phrases specifying, or describing more exactly, the target of a cross-reference, which form the content of the element. Since its content thus serves as a human-readable pointer, in the simplest case a <ref> element need not identify its target in any other way. For example:

```
See <ref>section 12 on page 34</ref>.
```

More usually, it will be desirable to identify the target of the cross-reference using the **target** attribute, so that processing software can access it directly, for example to implement a linkage, to generate an appropriate reference, or to give an error message if it cannot be found. Assuming that section 12 in the previous example has been tagged

```
<div1  xml:id="SEC12">
<!-- ... -->
</div1>
```

then the same cross-reference might more exactly be encoded as

```
See especially <ref  target="#SEC12">section 12 on page 34</ref>.
```

If the text for the cross-reference is to be generated according to a fixed pattern, or if no text is to appear in the body of the cross-reference, the <ptr> element would be used as follows:

```
See in particular <ptr  target="#SEC12"/>.
```

A cross-reference may point to any number of locations simultaneously, simply by giving more than one identifier as the value of its **target** attribute. This may be particularly useful where an analytic index is to be encoded, as in the following example:

```
<list>
  <item>Saints aid rejected in mel. <ptr  target="#p299"/>
  </item>
  <item>Sallets censured <ptr  target="#p143 #p144"/>
  </item>
  <item>Sanguine mel. signs <ptr  target="#p263"/>
  </item>
  <item>Scilla or sea onyon, a purger of mel. <ptr  target="#p442"/>
  </item>
</list>
```

Source: [24]

Here the targets of the cross-references are simply page numbers; it is assumed that corresponding elements with identifiers p299, p143, etc. have been provided in the body of the text, for example as page breaks

```
<pb  xml:id="p143"/>
...
<pb  xml:id="p144"/>
...
<pb  xml:id="p263"/>
...
<pb  xml:id="p299"/>
...
<pb  xml:id="p442"/>
...
```

The **type** attribute may be used, as elsewhere, to categorize the cross-reference according to any system of importance to the encoder. If bibliographic references require special processing (e.g. in order to provide a consistent short-form reference), they might be tagged thus:

```
Similar forms, often called
<term  rend="ldquo rdquo">rewriting systems</term>, have a long history
among mathematicians, but the specific form of <ptr  target="#fig22"/>
was first studied extensively by Chomsky <ptr  type="bibliog"  target="#chom59"/>.

<!-- ... -->
<figure  xml:id="fig22">
<!-- ... -->
```

```
</figure>
<!-- elsewhere, in the bibliography -->
<bibl xml:id="chom59">
<!-- citation for the book referenced above -->
</bibl>
```

Source: [93]

The value **bibliog** for the **type** attribute on the second <ptr> element here might be used to indicate that the object being referenced here is a bibliographic entry rather than a simple cross-reference to an illustration, as is the first <ptr>. In either case, the value of the **target** attribute is a pointer to some other element.

The <ptr> and <ref> elements have many applications in addition to the simple cross-referencing facilities illustrated in this section. In conjunction with the analytic tools discussed in chapters *16. Linking, Segmentation, and Alignment*, *17. Simple Analytic Mechanisms*, and *18. Feature Structures*, they may be used to link analyses of a text to their object, to combine corresponding segments of a text, or to align segments of a text with a temporal or other axis or with each other.

3.7 Lists

The following elements are provided for the encoding of lists, their constituent items, and the labels or headings associated with them:

<list> contains any sequence of items organized as a list.

<item> contains one component of a list.

<label> contains the label associated with an item in a list; in glossaries, marks the term being defined.

<head> (heading) contains any type of heading, for example the title of a section, or the heading of a list, glossary, manuscript description, etc.

<headLabel> (heading for list labels) contains the heading for the label or term column in a glossary list or similar structured list.

<headItem> (heading for list items) contains the heading for the item or gloss column in a glossary list or similar structured list.

The <list> element should be used to mark any kind of list: numbered, lettered, bulleted, or unmarked. Lists formatted as such in the copy text should in general be encoded using this element, with an appropriate value for the **type** attribute. Lists given as run-on text may also be encoded using this element, where this is felt to be appropriate.

Each distinct item in the list should be encoded as a distinct <item> element. If the numbering or other identification for the items in a list is unremarkable and may be reconstructed by any processing program, no enumerator need be specified. If however an enumerator is retained in the encoded text, it may be supplied either by using the **n** attribute on the <item> element, or by using a <label> element. The following examples are thus equivalent:

```
I will add two facts, which have seldom occurred in
the composition of six, or at least of five quartos.
<list rend="runon" type="ordered">
  <label>(1)</label>
  <item>My first rough manuscript, without any
     intermediate copy, has been sent to the press.</item>
  <label>(2)</label>
  <item>Not a sheet has been seen by any human
     eyes, excepting those of the author and the printer:
```

```
    the faults and the merits are exclusively my own.</item>
</list>
```

```
I will add two facts, which have seldom occurred in
the composition of six, or at least of five quartos.
<list  rend="runon"  type="ordered">
  <item  n="1">My first rough manuscript, without any
    intermediate copy, has been sent to the press.</item>
  <item  n="2">Not a sheet has been seen by any human
    eyes, excepting those of the author and the printer:
    the faults and the merits are exclusively my own.</item>
</list>
```

The two styles may not be mixed in the same list: if one item is preceded by a label, all must be.

A list need not necessarily be displayed in list format. For example, the following is a reasonable encoding of a list which (in the original) is simply printed as a single paragraph:

```
On those remote pages it is written that animals are
divided into <list>
  <item  n="a">those that belong to the Emperor, </item>
  <item  n="b">embalmed ones, </item>
  <item  n="c">those that are trained, </item>
  <item  n="d">suckling pigs, </item>
  <item  n="e">mermaids, </item>
  <item  n="f">fabulous ones, </item>
  <item  n="g">stray dogs, </item>
  <item  n="h">those that are included in this classification, </item>
  <item  n="i">those that tremble as if they were mad, </item>
  <item  n="j">innumerable ones, </item>
  <item  n="k">those drawn with a very fine camel's-hair brush, </item>
  <item  n="l">others, </item>
  <item  n="m">those that have just broken a flower vase, </item>
  <item  n="n">those that resemble flies from a distance. </item>
</list>
```

Source: [18]

A list may be given a heading or title, for which the <head> element should be used, as in the next example, which also demonstrates simple use of the <label> element to mark a tabular or glossary list in which each item is associated with a word or phrase rather than a numeric or alphabetic enumerator:

```
<list  type="gloss">
  <head>Report of the conduct and progress of Ernest Pontifex.
    Upper Vth form — half term ending Midsummer 1851</head>
  <label>Classics</label>
  <item>Idle listless and unimproving</item>
  <label>Mathematics</label>
  <item>ditto</item>
  <label>Divinity</label>
  <item>ditto</item>
  <label>Conduct in house</label>
  <item>Orderly</item>
```

```
  <label>General conduct</label>
  <item>Not satisfactory, on account of his great
     unpunctuality and inattention to duties</item>
</list>
```

Source: [25]

In such a list, the individual items have internal structure. In complex cases, where list items contain many components, the list is better treated as a table, on which see chapter *14. Tables, Formulæ, and Graphics.* A particularly important instance of the simple two-column table is the 'glossary list', which should be marked by the tag **<list type="gloss">**. In such lists, each <label> element contains a term and each <item> its gloss; it is a semantic error for a list tagged with `type="gloss"` not to have labels. For example:

```
<list  type="gloss">
  <head>Unit Three — Vocabulary</head>
  <label  xml:lang="la">acerbus, -a, -um </label>
  <item>bitter, harsh</item>
  <label  xml:lang="la">ager, agrī, M. </label>
  <item>field</item>
  <label  xml:lang="la">audiō, īre,
     īvī, ītus </label>
  <item>hear, listen (to)</item>
  <label  xml:lang="la">bellum, -ī, N. </label>
  <item>war</item>
  <label  xml:lang="la">bonus, -a, -um </label>
  <item>good</item>
</list>
```

Source: [146]

Additionally, the <term> and <gloss> elements discussed in section *3.3.4. Terms, Glosses, Equivalents, and Descriptions* might be used to make explicit the role that each column in the glossary list has, as follows:

```
<list  type="gloss">
  <head>Unit Three — Vocabulary</head>
  <label>
    <term  xml:lang="la">acerbus, -a, -um</term>
  </label>
  <item>
    <gloss>bitter, harsh</gloss>
  </item>
  <label>
    <term  xml:lang="la">ager, agrī, M. </term>
  </label>
  <item>
    <gloss>field</gloss>
  </item>
  <label>
    <term  xml:lang="la">audiō, -īre, -īvī, -ītus</term>
  </label>
  <item>
    <gloss>hear, listen (to)</gloss>
  </item>
  <label>
    <term  xml:lang="la">bellum, -ī, N. </term>
```

```
  </label>
  <item>
    <gloss>war</gloss>
  </item>
  <label>
    <term  xml:lang="la">bonus, -a, -um</term>
  </label>
  <item>
    <gloss>good</gloss>
  </item>
</list>
```

Note in the above examples the use of the global **xml:lang** attribute to specify on the <label> (or <term>) element what language the term is from. For further discussion of the **xml:lang** attribute see section *1.3.1.1. Global Attributes*, and section *vi.1 Language identification*. A more elaborate markup for this glossary would distinguish the headword forms from the grammatical information (principal parts and gender), perhaps using elements taken from *9. Dictionaries*.

In addition to the <head> element used to supply a title or heading for the whole list, headings for the two columns of a glossary-style list may be specified using the two special elements <headLabel> and <headItem>:

```
The simple, straightforward statement of an idea is
preferable to the use of a worn-out expression.
<list  type="gloss">
  <headLabel>TRITE</headLabel>
  <headItem>SIMPLE, STRAIGHTFORWARD</headItem>
  <label>bury the hatchet </label>
  <item>stop fighting, make peace</item>
  <label>at loose ends </label>
  <item>disorganized</item>
  <label>on speaking terms </label>
  <item>friendly</item>
  <label>fair and square </label>
  <item>completely honest</item>
  <label>at death's door </label>
  <item>near death</item>
</list>
```

Source: [206]

The elements <label>, <head>, <headLabel>, and <headItem> may contain only phrase-level elements. The <item> element however may contain paragraphs or other 'chunks', including other lists. In this example, a glossary list contains two items, each of which is itself a simple list:

```
<list  type="gloss">
  <label>EVIL</label>
  <item>
    <list  type="simple">
      <item>I am cast upon a horrible desolate island, void
            of all hope of recovery.</item>
      <item>I am singled out and separated as it were from
            all the world to be miserable.</item>
      <item>I am divided from mankind — a solitaire; one
            banished from human society.</item>
```

```
    </list>
  </item>
  <label>GOOD</label>
  <item>
    <list  type="simple">
      <item>But I am alive; and not drowned, as all my
            ship's company were.</item>
      <item>But I am singled out, too, from all the ship's
            crew, to be spared from death...</item>
      <item>But I am not starved, and perishing on a barren place,
            affording no sustenances....</item>
    </list>
  </item>
</list>
```

Source: [54]

Lists of different types may be nested to arbitrary depths in this way.

3.8 Notes, Annotation, and Indexing

3.8.1 Notes and Simple Annotation

The following elements are provided for the encoding of discursive notes, whether already present in the copy text or supplied by the encoder:

<note> contains a note or annotation.

A note is any additional comment found in a text, marked in some way as being out of the main textual stream. All notes should be marked using the same tag, <note>, whether they appear as block notes in the main text area, at the foot of the page, at the end of the chapter or volume, in the margin, or in some other place.

Notes may be in a different hand or typeface, may be authorial or editorial, and may have been added later. Attributes may be used to specify these and other characteristics of notes, as detailed below.

Where possible, the body of a note should be inserted in the text at the point at which its identifier or mark first appears. This may not be possible for example with marginal notes, which may not be anchored to an exact location. For simplicity, it may be adequate to position marginal notes before the relevant paragraph or other element. In some cases, however, it may be desirable to transcribe notes not at their point of attachment to the text but at their point of appearance (at the end of the volume, or the end of the chapter — not, in general, when the notes appear at the foot of the page); in this case the **target** attribute should be used to specify the point of attachment. In some cases, the note is explicitly attached not to a point but to a span of text; in which case both the **target** and **targetEnd** attributes should be used to specify the span of attachment. For a full discussion of pointing to points and spans in the text, see section *3.6. Simple Links and Cross-References*.

Examples:

```
<l>The self-same moment I could pray</l>
<l>And from my neck so free</l>
<l>The albatross fell off, and sank</l>
<l>Like lead into the sea.
<note  type="auth"  place="margin">The spell begins to break</note>
</l>
```

Source: [42]

```
Collections are ensembles of distinct entities or objects
of any sort.<note  n="1"  place="foot">We explain below why we use
the uncommon term <mentioned>collection</mentioned>
instead of the expected <mentioned>set</mentioned>.
Our usage corresponds to the <mentioned>aggregate</mentioned> of many
mathematical writings and to the sense of <mentioned>class</mentioned>
found in older logical writings.</note> The elements ...
```

In addition to transcribing notes already present in the copy text, researchers may wish to add their own notes or comments to it. The <note> element may be used for either purpose, but it will usually be advisable to distinguish the two categories, for example by means of the **type** or **resp** attributes. When annotating the electronic text by means of analytic notes in some structured vocabulary, e.g. to specify the topics or themes of a text, the and <interp> elements may be preferable; these elements are available when the module for simple analysis is selected (see section *17.3. Spans and Interpretations*).

3.8.2 Index Entries

The indexing of scholarly texts is a skilled activity, involving substantial amounts of human judgment and analysis. It should not therefore be assumed that simple searching and information retrieval software will be able to meet all the needs addressed by a well-crafted manual index, although it may complement them for example by providing free text search. The role of an index is to provide access via keywords and phrases which are not necessarily present in the text itself, but must be added by the skill of the indexer.

3.8.2.1 Pre-existing indexes

When encoding a pre-existing text, therefore, if such an index is present it may be advisable to retain it along with the text, rather than attempt to regenerate it automatically. Elements discussed elsewhere in these Guidelines may be used for this purpose. For example, the <div1> element or element may be used to mark the section of the text containing the index and the <list> element might be used to mark the index itself, each entry being represented by an <item> element, possibly containing within it a series of <ptr> or <ref> elements, as follows:

```
<div  type="index">
<!--...-->
  <list  type="index">
    <item>Women, how cause of mel. <ref>193</ref>; their vanity in
        apparell taxed, <ref>527</ref>; their counterfeit tears
    <ref>547</ref>; their vices <ref>601</ref>, commended,
    <ref>624</ref>.</item>
    <item>Wormwood, good against mel. <ref>443</ref>
    </item>
    <item>World taxed, <ref>181</ref>
    </item>
    <item>Writers of the cure of mel. 295</item>
<!--...-->
  </list>
</div>
```

Source: [24]

Note that this simple representation does not capture the nested structure of the first of these index entries. A more accurate representation might entail the use of nested lists like the following:

```
<item>Women,
<list>
   <item>how cause of mel. <ref>193</ref>;</item>
   <item>their vanity in apparell taxed, <ref>527</ref>;</item>
   <item>their counterfeit tears <ref>547</ref>;</item>
   <item>their vices
   <list>
      <item>
        <ref>601</ref>,</item>
      <item> commended, <ref>624</ref>.</item>
    </list>
   </item>
  </list>
</item>
```

The page references, encoded simply as <ref> elements above, might also include direct links to the appropriate location in the encoded text, using (for example) a target attribute to supply the identifier of an associated page break element:

```
<!-- in the text --><pb  xml:id="P624"/>
<!-- start of page 624 -->
<!-- in the index -->
<ref  target="#P624">624</ref>
```

For further discussion of this and alternative ways of encoding such links see the discussion in section *16. Linking, Segmentation, and Alignment.* Note that similar methods may also be used to encode a table of contents, as further exemplified in section *4.5. Front Matter*.

3.8.2.2 Auto-generated indexes

It can also be useful, however, to generate a new index from a machine-readable text, whether the text is being written for the first time with the tags here defined, or as an addition to a text transcribed from some other source. Depending on the complexity of the text and its subject matter, such an automatically-generated index may not in itself satisfy all the needs of scholarly users. However it can assist a professional indexer to construct a fully adequate index, which might then be post-edited into the digital text, marked-up along the lines already suggested for preserving pre-existing index material.

Indexes generally contain both references to specific pages or sections and references to page ranges or sequences. The same element is used in either case:

<index> (index entry) marks a location to be indexed for whatever purpose.

Like the <interp> element described in *17.3. Spans and Interpretations* this element may be used simply to provide descriptive or interpretive label of some kind for any location within a text, to be processed in any way by analytic software, but its main purpose is to facilitate the generation of an index for a printed version of the text. An <index> element may be placed anywhere within a text, between or within other elements. The headwords to be used when making up this index are given by the <term> elements within the <index> element. The location of the generated index might be specified by means of a processing instruction within the text, such as the following (the exact form of the PI is of course dependent on the application software in use):

```
    <?tei indexplacement ?>
```

Alternatively, the special purpose <divGen> element might be used.

In the simplest case, a single headword is supplied by an <term> elements contained by an <index> element:

```
<p>The students understand procedures for Arabic lemmatisation
<index>
   <term>Lemmatization, Arabic</term>
  </index>and are beginning to build parsers.</p>
```

The effect of this will be to generate an index entry for the term 'Lemmatization', referencing the location of the original <index> element.

If the subject of Arabic lemmatization is treated at length in a text, then the index entry generated may need to reference a sequence of locations (e.g. page numbers). In such a case it will be necessary to identify the end of the relevant span of text as well as its starting point. This is most conveniently done by supplying an empty <anchor> element (as discussed in chapter *16. Linking, Segmentation, and Alignment*) at the appropriate point and pointing to it from the <index> element by means of its **spanTo** attribute, as in this example:

```
<p>We now turn to the
topic of Arabic lemmatisation
<index  spanTo="#ALAMEND">
   <term>Lemmatization, Arabic</term>
  </index> concerning which it is important to note .....

<!-- much learned material omitted here -->
and now we can build our parser.<anchor  xml:id="ALAMEND"/>
</p>
```

This would generate the same index entries as the previous example, but the reference would be to the whole span of text between the location of the <index> element and the location of the element identified by the code ALAMEND, rather than a single point, and thus might (for example) include a sequence of page numbers.

Although the position of the <index> element in the text provides the target location that will be specified in the generated index entry, no part of the text itself is used to construct that entry. Index terms appearing in the entry come solely from the content of <term> elements, which consequently may have to repeat words or phrases from the text proper. This need not be done verbatim, thus giving scope for normalization of spelling (as in the example above) or other modifications which may assist generation of an index in a desired form or sequence.

Sometimes, for example when index terms are taken from a different language or consist of mathematical formulae or other expressions, even a normalized form of an index term may be insufficent for an application to order it exactly as desired. The **sortKey** attribute may be used to address this problem, as in the following example:

```
<p>The @ operator
<index>
   <term  sortKey="0000">@</term>
  </index> precedes an
attribute name</p>
```

Here, an entry for the symbol @ will appear in the index, but will be sorted alphabetically as if it were the string **0000**. This technique is also useful when an index entry is to contain some non-Unicode character or glyph represented by the <g> element discussed in chapter *5. Representation of Non-standard Characters and*

Glyphs. In the following example, we assume that somewhere a definition for this glyph has been provided using the elements described in chapter *5. Representation of Non-standard Characters and Glyphs*, and given the code **PrinceGlyph**:

```
<char  xml:id="PrinceGlyph">
<!-- definition of the glyph here -->
</char>
<p>The Artist formerly known as Prince <index>
   <term  sortKey="Prince">
     <g  ref="#PrinceGlyph"/>
   </term>
  </index>...</p>
```

Note that if no value is supplied for the sortKey attribute, a sorting application should always use the content of the <term> element as a sort key.

It is common practice to compile more than one index for a given text. A biography of a poet, for example, may offer an index of references to poems by the subject of the study, another index of works by other writers, an index of places or historical personages etc. The indexName attribute is used to assigning index terms and locations to one or more specific indexes:

```
<p>Sir John Ashford
<index  indexName="INDEX-PERSONS">
   <term>Ashford, John</term>
  </index> was,
coincidentally, born in
<index  indexName="INDEX-PLACES">
   <term>Ashford
       (Kent)</term>
  </index>Ashford...</p>
```

Multi-level indexing is particularly common in scholarly documents. For example, as well as entries such as TEI, or markup, an index may contain structured entries like TEI, markup practices, index terms, where a top level entry TEI is followed by a number of second-level subcategories, any or all of which may have a third-level list attached to them and so on. In order to reflect such a hierarchical index listing, <index> elements may be nested to the required depth. For example, suppose that we wish to make a structured index entry for 'lemmatisation' with subentries for 'Arabic', 'Sanskrit', etc. The example at the start of this section might then be encoded with nested <index> elements:

```
<p>The students understand procedures for Arabic lemmatisation
<index>
   <term>lemmatization</term>
   <index>
     <term>arabic</term>
   </index>
  </index>
...</p>
```

The index entry from Burton's *Anatomy of Melancholy* quoted above might be generated in a similar way. To generate such an entry, the body of the text might include, at page 193, an <index> element such as

```
<index>
  <term>Women</term>
  <index>
    <term>how cause of mel.</term>
  </index>
</index>
```

. Similary, page 601 of the body text would include an <index> element like the following:

```
<index>
  <term>Women</term>
  <index>
    <term>their vices</term>
  </index>
</index>
```

while the <index> element at page 624 would have a structure like the following:

```
<index>
  <term>Women</term>
  <index>
    <term>their vices</term>
    <index>
      <term>commended</term>
    </index>
  </index>
</index>
```

When processing such <index> elements, the duplication required to make the structure explicit will normally be removed, so as to produce entries like those quoted above. However, this is not required by the encoding recommended here.

As noted above, either a processing instruction or a <divGen> element may be used to mark the place at which an index generated from <index> elements should be inserted into the output of a processing program; typically but not necessarily this will be at some point within the back matter of the document. If the <divGen> element is used, then the **type** attribute should be used to specify which kind of index is to be generated, and its value should correspond with that of the **indexName** attribute on the relevant <index> elements.

```
<back>
  <div  type="appendix">
    <head>Bibliography</head>
    <listBibl>
      <bibl> ... </bibl>
    </listBibl>
  </div>
  <divGen  n="Index Nominum"  type="INDEX-NAMES"/>
  <divGen  n="Index Loci"  type="INDEX-PLACES"/>
</back>
```

As this example shows, the global **n** attribute may also be used to specify a name or identifier for the generated index itself in the usual way. Any additional headings etc. required for the generated index must be specified as content of the <divGen> element.

```
<back>
  <divGen  n="A1"  type="INDEX-NAMES">
    <head>An Index of Names</head>
  </divGen>
</back>
```

If a processing instruction is used, then these parameters for the generated index may be supplied in some other way.

One final feature frequently found in manually-created indexes to printed works cannot readily be encoded by the means provided here, namely cross-references internal to the index term listing. For example, if all references to the TEI in a text have been indexed using the index term Text Encoding Initiative, it may also be helpful to include an entry under the term TEI containing some text such as 'see Text Encoding Initiative'. Such internal cross-references must be added as part of the post-editing phase for an auto-generated index.

3.9 Graphics and other non-textual components

Graphics, such as illustrations or diagrams, appear in many different kinds of text, and often with different purposes. In some cases, the graphic is an integral part of a text (indeed, some texts — comic books for example — may be almost entirely graphic); in others the graphic may be a kind of optional extra. In some cases, the text may be incomprehensible unless the graphic is included; in others, the presence of the graphic adds very little to the sense of the work. It will therefore be a matter of encoding policy as to whether or how a graphic found in a source text is transferred to a digital version of the same. In documents which are 'born digital', graphics and other forms of non-textual element may be particularly salient, but their inclusion in an archival form of the document concerned remains an editorial decision.

Considered as structural components, graphics may be anchored to a particular point in the text, or they may float either completely freely, or within some defined scope, such as a chapter or section. Graphics of this kind often contain associated text such as a heading or label, and may also nest hierarchically. These Guidelines recommend the following different elements for these two cases:

<figure> groups elements representing or containing graphic information such as an illustration or figure.

<graphic/> indicates the location of an inline graphic, illustration, or figure.

<binaryObject> provides encoded binary data representing an inline graphic or other object.

Graphic components may be encoded in a number of different ways:

- in some non-XML or binary format such as PNG, JPEG, etc.
- in an XML format such as SVG
- in a TEI XML format such as the notation for graphs and trees described in *19. Graphs, Networks, and Trees*

In the last two cases, the presence of the graphic will be indicated by an appropriate XML element, drawn from the SVG namespace in the second case, and its content will fully define the graphic to be produced. In the first case, the element <graphic> is used to mark the presence of the graphic only and the visual content is stored outside the XML document, and its location is referenced by means of an **url** attribute. Alternatively, if the graphical information is embedded directly within the document using some suitable binary format such as Base64, the <binaryObject> element may be used to contain it.

The elements <graphic> and <binaryObject> are made available as members of the class model.graphicLike when this module is included in a schema. These elements are also both members of the class att.internetMedia, from which they inherit the following attribute:

att.internetMedia provides attributes for specifying the type of a computer resource using a standard taxonomy.

@**mimeType** (MIME media type) specifies the applicable multimedia internet mail extension (MIME) media type

For example, the following passage indicates that a copy of the image found in the source text may be recovered from the URL zigzag2.png and that this image is in PNG format:

```
<p>These were the four lines I moved in
through my first, second, third, and
fourth volumes. -- In the fifth volume
I have been very good, -- the precise
line I have described in it being this :
<graphic url="zigzag2.png" mimeType="image/png"/>
By which it appears, that except at the
curve, marked A. where I took a trip
to Navarre, -- and the indented curve B.
which is the short airing when I was
there with the Lady Baussiere and her
page, -- I have not taken the least frisk
...</p>
```

Source: [187]

The <graphic> and <binaryObject> elements are phrase level elements which may be used anywhere that textual content is permitted, within but not between paragraphs or headings. In the following example, the encoder has decided to treat a specific printer's ornament as a heading:

```
<head>
  <graphic
     url="http://www.iath.virginia.edu/gants/Ornaments/Heads/hp-ral02.gif"/>
</head>
```

.

The <figure> element discussed in *14.3. Specific Elements for Graphic Images* provides additional capabilities, for example the ability to combine a number of images into a hierarchically organized structure or a block of images. It also provides the ability to associate an image with additional information such as a heading or a description.

3.10 Reference Systems

By *reference system* we mean the system by which names or references are associated with particular passages of a text (e.g. *Ps. 23:3* for the third verse of Psalm 23 or *Amores 2.10.7* for Ovid's *Amores*, book 2, poem 10, line 7). Such names make it possible to mark a place within a text and enable other readers to find it again. A reference system may be based on structural units (chapters, paragraphs, sentences; stanza and verse), typographic units (page and line numbers), or divisions created specifically for reference purposes (chapter and verse in Biblical texts). Where one exists, the traditional reference system for a text should be preserved in an electronic transcript of it, if only to make it easier to compare electronic and non-electronic versions of the text.

Reference systems may be recorded in TEI-encoded texts in any of the following ways:

- where a reference system exists, and is based on the same logical structure as that of the text's markup, the reference for a passage may be recorded as the value of the global **xml:id** or **n** attribute on an appropriate tag, or may be constructed by combining attribute values from several levels of tags, as described below in section *3.10.1. Using the xml:id and n Attributes*.

- where there is no pre-existing reference system, the global **xml:id** or **n** attributes may be used to construct one (e.g. collections and corpora created in electronic form), as described below in section *3.10.2. Creating New Reference Systems*.
- where a reference system exists which is not based on the same logical structure as that of the text's markup (for example, one based on the page and line numbers of particular editions of the text rather than on the structural divisions of it), any of a variety of methods for encoding the logical structure representing the reference system may be employed, as described in chapter *20. Non-hierarchical Structures*.
- where a reference system exists which does not correspond to any particular logical structure, or where the logical structure concerned is of no interest to the encoder except as a means of supporting the referencing system, then references may be encoded by means of <milestone> elements, which simply mark points in the text at which values in the reference system change, as described below in section *3.10.3. Milestone Elements*.

The specific method used to record traditional or new reference systems for a text should be declared in the TEI header, as further described in section *3.10.4. Declaring Reference Systems* and in section *16.2.5. Canonical References*.

When a text has no pre-existing associated reference system of any kind, these Guidelines recommend as a minimum that at least the page boundaries of the source text be marked using one of the methods outlined in this section. Retaining page breaks in the markup is also recommended for texts which have a detailed reference system of their own. Line breaks in prose texts may be, but need not be, tagged.[5]

3.10.1 Using the **xml:id** and **n** Attributes

When traditional reference schemes represent a hierarchical structuring of the text which mirrors that of the marked-up document, the **n** attribute defined for all elements may be used to indicate the traditional identifier of the relevant structural units. The **n** attribute may also be used to record the numbering of sections or list items in the copy text if the copy-text numbering is important for some reason, for example because the numbers are out of sequence.

For example, a traditional reference to Ovid's *Amores* might be *Amores 2.10.7*—book 2, poem 10, line 7. Book, poem, and line are structural units of the work and will therefore be tagged in any case. (See chapter *6. Verse* for a discussion of structural units in verse collections.) In such cases, it is convenient to record traditional reference numbers of the structural units using the **n** attribute. The relevant tags for our example would be:

```
<div1 n="Amores" type="volume">
 <div2 n="1" type="book">
<!-- ... -->
 </div2>
 <div2 n="2" type="book">
  <div3 n="1" type="poem">
<!-- ... -->
  </div3>
  <div3 n="2" type="poem">
<!-- ... -->
  </div3>
<!-- ... -->
  <div3 n="10" type="poem">
    <l n="1"> ... </l>
    <l n="2"> ... </l>
<!-- ... -->
```

[5] Many encoders find it convenient to retain the line breaks of the original during data entry, to simplify proofreading, but this may be done without inserting a tag for each line break of the original.

```
     <l  n="7"> ... </l>
   </div3>
<!-- ... -->
  </div2>
<!-- ... -->
</div1>
```

One may also place the entire standard reference for each portion of the text into the appropriate value for the **n** attribute, though for obvious reasons this takes more space in the file:

```
<div1  n="Amores"  type="volume">
  <div2  n="Amores 1"  type="book">
<!-- ... -->
  </div2>
  <div2  n="Amores 2"  type="book">
    <div3  n="Amores 2.1"  type="poem">
<!-- ... -->
    </div3>
<!-- ... -->
    <div3  n="Amores 2.10"  type="poem">
<!-- ... -->
     <l  n="Amores 2.10.7"> ... </l>
<!-- ... -->
    </div3>
<!-- ... -->
  </div2>
<!-- ... -->
</div1>
```

If the names used by the traditional reference system can be formulated as identifiers, then the references can be given as values for the **xml:id** attribute; this requires that the reference be given without internal spaces, begin with a letter or underscore, and contain no characters other than letters, digits, hyphens, underscores, full stops, and the various combining and extender characters, as defined by the XML specification. Unlike values for the **n** attribute, values for the **xml:id** attribute must be unique throughout the document. Our example then looks like this:

```
<div1  n="Amores"  type="volume">
  <div2  xml:id="am.1"  type="book">
<!-- ... -->
  </div2>
  <div2  xml:id="am.2"  type="book">
    <div3  xml:id="am.2.1"  type="poem">
<!-- ... -->
    </div3>
<!-- ... -->
    <div3  xml:id="am.2.10"  type="poem">
<!-- ... -->
     <l  xml:id="am.2.10.7"> ... </l>
<!-- ... -->
    </div3>
<!-- ... -->
  </div2>
<!-- ... -->
</div1>
```

To document the usage and to allow automatic processing of these standard references, it is recommended that the TEI header be used to declare whether standard references are recorded in the **n** or **xml:id** attributes and which elements may carry standard references or portions of them. For examples of declarations for the reference systems just shown, see section *3.10.4. Declaring Reference Systems*.

Using the **n** attribute one can specify only a single standard referencing system, a limitation not without problems, since some editions may define structural units differently and thus create alternative reference systems. For example, another edition of the *Amores* considers poem 10 a continuation of poem 9, and therefore would specify the same line as *Amores 2.9.31*. In order to record both of these reference systems one could employ any of a variety of methods discussed in chapter *20. Non-hierarchical Structures*.

3.10.2 Creating New Reference Systems

If a text has no canonical reference system of its own, a reference system, if needed, may be derived from the structure of the electronic text, specifically from the markup of the text. As with any reference system intended for long-term use, it is important to see the reference as an established, unchanging point in the text. Should the text be revised or rearranged, the reference-system identifiers associated with any bit of text must stay with that bit of text, even if it means the reference numbers fall out of sequence. (A new reference system may always be created beside the old one if out-of-sequence numbers must be avoided.)

The global attributes **n** and **xml:id** may be used to assign reference identifiers to segments of the text. Identifiers specified by either attribute apply to the entire element for which they are given. ID attributes must be unique within a single document, and ID values must begin with a letter. No such restrictions are made on the values of **n** attributes.

A convenient method of mechanically generating unique values for **xml:id** or **n** attributes based on the structure of the document is to construct, for each element, a domain-style address comprising a series of components separated by full stops, with one component for each level of the document hierarchy. Two methods may be used. In the typed path form of identifier, each component in the identifier takes the form of an element identifier, a hyphen, and a number, for example `p-2`. The element name specifies what type of element is to be sought, and the number specifies which occurrence of that element type is to be selected. (The hyphen and number may be omitted if there is only one element of the given type.) In the untyped path form of identifier, each component consists of a number, indicating which element in the sequence of nodes at each level is to be selected.

Identifiers generated with these methods should use the <text> element as their starting point, rather than the <TEI> or <body> elements. The <TEI> element may be taken as a starting point only if identifiers need to be generated for the <teiHeader>, which is not usually the case; using the <body> element as a root would prevent assignment of identifiers for the front and back matter. The component corresponding to the root element can be omitted from identifiers, if no confusion will result. In collections and corpora, the component corresponding to the root may be replaced by the unique identifier assigned to the text or sample.

In the following example, each element within the <text> element has been given a typed-path identifier as its **xml:id** value, and an untyped-path identifier as its **n** value; the latter are prefixed with the string *AB*, which may be imagined to be the general identifier for this text.

```
<text xml:id="Text-1" n="AB">
  <front xml:id="Front" n="AB.1">
    <div xml:id="Front.div-1" n="AB.1.1">
      <p> ... </p>
    </div>
    <titlePage xml:id="Front.titlePage" n="AB.1.2">
      <titlePart> ... </titlePart>
    </titlePage>
```

```
    <div xml:id="Front.div-2" n="AB.1.3">
      <p> ... </p>
    </div>
  </front>
  <body xml:id="Body" n="AB.2">
    <p xml:id="Body.p-1" n="AB.2.1"> ... </p>
    <p xml:id="Body.p-2" n="AB.2.2"> ... </p>
    <div xml:id="Body.div-1" n="AB.2.3">
      <head xml:id="Body.div-1.head" n="AB.2.3.1"> ... </head>
      <p xml:id="Body.div-1.p-1" n="AB.2.3.2"> ... </p>
      <p xml:id="Body.div-1.p-2" n="AB.2.3.3"> ... </p>
    </div>
    <div xml:id="Body.div-2" n="AB.2.4">
      <head xml:id="Body.div-2.head" n="AB.2.4.1"> ... </head>
      <p xml:id="Body.div-2.p-1" n="AB.2.4.2"> ... </p>
      <p xml:id="Body.div-2.p-2" n="AB.2.4.3"> ... </p>
    </div>
  </body>
</text>
```

The typed and untyped path methods are convenient, but are in no way required for anyone creating a reference system.

If the **xml:id** attribute is used to record the reference identifiers generated, each value should record the entire path. If the **n** attribute is used, each value may record either the entire path or only the subpath from the parent element. The attribute used, the elements which can bear standard reference identifiers, and the method for constructing standard reference identifiers, should all be declared in the header as described in section *2.3.5. The Reference System Declaration*.

3.10.3 Milestone Elements

Where the desired reference system does not correspond to any particular structural hierarchy, or the document combines multiple structural hierarchies (as further discussed in *20. Non-hierarchical Structures*), simpler though less expressive methods may be necessary. In such cases the simplest solution may be just to mark up changes in the reference system where they occur, by using one or more of the following milestone elements:

<milestone/> marks a boundary point separating any kind of section of a text, typically but not necessarily indicating a point at which some part of a standard reference system changes, where the change is not represented by a structural element.

<pb/> (page break) marks the boundary between one page of a text and the next in a standard reference system.

<lb/> (line break) marks the start of a new (typographic) line in some edition or version of a text.

<cb/> (column break) marks the boundary between one column of a text and the next in a standard reference system.

These elements simply mark the points in a text at which some category in a reference system changes. They have no content but subdivide the text into regions, rather in the same way as milestones mark points along a road, thus implicitly dividing it into segments. The elements <pb>, <cb>, and <lb> are specialised types of milestone, marking page, column, and line boundaries. The global **n** attribute is used in each case to provide a value for the particular unit associated with this milestone (for example, the page or line number). Since it is not structural, validation of a reference system based on <milestone>s cannot be checked by an XML parser, so it will be the responsibility of the encoder or the application software to ensure that they are given in the correct order.

Milestones are useful where a text has two competing structures. For example, many English novels were first published as serial works, individual parts of which do not always contain a whole number of chapters. An encoder might decide to represent the chapter-based structure using <div1> elements, with <milestone> elements to mark the points at which individual parts end; or the reverse. Thus, an encoding in which chapters are regarded as more important than parts might encode some work in which chapter three begins in part one and is concluded in part two as follows:

```
<text>
  <body>
    <milestone unit="part" n="1"/>
    <div1 n="1" type="chapter">
      <p>
<!-- ... -->
      </p>
    </div1>
    <div1 n="2" type="chapter">
      <p>
<!-- ... -->
      </p>
    </div1>
    <div1 n="3" type="chapter">
      <p>
<!-- ... -->
      </p>
      <milestone unit="part" n="2"/>
      <p>
<!-- ... -->
      </p>
    </div1>
  </body>
</text>
```

An encoding of the same work in which parts are regarded as more important than chapters might begin as follows:

```
<text>
  <body>
    <div1 n="1" type="part">
      <milestone unit="chapter" n="1"/>
      <p>
<!-- ... -->
      </p>
      <milestone unit="chapter" n="2"/>
      <p>
<!-- ... -->
      </p>
      <milestone unit="chapter" n="3"/>
      <p>
<!-- ... -->
      </p>
    </div1>
    <div1 n="2" type="part">
      <p>
<!-- ... -->
      </p>
```

```
     <milestone  unit="chapter"  n="4"/>
     <p>
<!-- ...  -->
     </p>
   </div1>
  </body>
</text>
```

Similarly, when tagging dramatic verse one may wish to privilege stanzas and lines over speeches and speakers, particularly where speeches cross line and line group boundaries. One might also wish to mark changes in narrative voice in a prose text. In either case, a milestone tag may be used to indicate change of speaker:

```
<lg>
  <milestone  unit="speaker"  n="Man"/>
  <l>Oh what is this I cannot see</l>
  <l>With icy hands gets a hold on me</l>
  <milestone  unit="speaker"  n="Death"/>
  <l>Oh I am Death, none can excel</l>
  <l>I open the doors of heaven and hell</l>
</lg>
```

Source: [33]

Milestone tags also make it possible to record the reference systems used in a number of different editions of the same work. The reference system of any one edition can be recreated from a text in which all are marked by simply ignoring all elements that do not specify that edition on their **ed** attribute.

As a simple example, assuming that edition E1 of some collection of poems regards the first two poems as constituting the first book, while edition E2 regards the first poem as prefatory, a markup scheme like the following might be adopted:

```
<milestone  ed="E1"  unit="work"/>
<milestone  ed="E2"  unit="work"/>
<milestone  ed="E1"  unit="book"/>
<milestone  ed="E1"  unit="poem"/>
<milestone  ed="E2"  unit="poem"/>
<milestone  ed="E2"  unit="book"/>
<milestone  ed="E1"  unit="poem"/>
<milestone  ed="E2"  unit="poem"/>
```

In this case no **n** value is specified, since the numbers rise predictably and the application can keep a count from the start of the document, if desired.

The value of the **n** attribute may but need not include the identifiers used for any larger sections. That is, either of the following styles is legitimate:

```
<milestone  ed="E1"  unit="work"  n="Amores"/>
<milestone  ed="E1"  unit="book"  n="1"/>
<milestone  ed="E1"  unit="poem"  n="1"/>
<milestone  ed="E1"  unit="poem"  n="2"/>
<milestone  ed="E1"  unit="book"  n="2"/>
```

or

```
<milestone ed="E1" unit="work" n="Amores"/>
<milestone ed="E1" unit="book" n="1"/>
<milestone ed="E1" unit="poem" n="1.1"/>
<milestone ed="E1" unit="poem" n="1.2"/>
<milestone ed="E1" unit="book" n="2"/>
```

When using <milestone> tags, line numbers may be supplied for every line or only periodically (every fifth, every tenth line). The latter may be simpler; the former is more reliable.

The style of numbering used in the values of **n** is unrestricted: for the example above, **I.i**, **I.ii**, and **I.iii** could have been used equally well if preferred. The special value **unnumbered** should be reserved for marking sections of text which fall outside the normal numbering system (e.g. chapter heads, poem numbers, titles, or speaker attributions in a verse drama).

By default, there are no constraints on the values supplied for the **ed** attribute. If it is felt appropriate to enforce such a restriction, the techniques described in *23.2. Personalization and Customization* may be used, for example to specify that the attribute must specify one of a predefined set of values.

See below, section *3.10.4. Declaring Reference Systems*, for examples of declarations for the reference systems just shown.

Milestone elements may be used to mark any kind of shift in the properties associated with a piece of text, even if this would not normally be considered a reference system. For example, they may be used to mark changes in narrative voice in a prose text, or changes of speaker in a dramatic text, where these are not marked using structural elements such as <sp>, perhaps in order to avoid a clash of hierarchies.

3.10.4 Declaring Reference Systems

Whatever kind of reference system is used in an electronic text, it is recommended that the TEI header contain a description of its construction in the <refsDecl> element described in section *2.3.5. The Reference System Declaration*. As described there, the declaration may consist either of a formal declaration using the <cRefPattern> element or an informal description in prose. The former is recommended because unlike prose it can be processed by software.

The three examples given in section *3.10.1. Using the xml:id and n Attributes* would be declared as follows. The first example encodes the standard references for Ovid's *Amores* one level at a time, using the **n** attribute on the <div1>, <div2>, <div3>, and <l> tags. The header for such an encoding should look something like this:

```
<teiHeader>
  <fileDesc>
<!-- ... -->
  </fileDesc>
  <encodingDesc>
    <refsDecl>
      <cRefPattern
         matchPattern="([^ ]+) ([0-9]+)\.([0-9]+)\.([0-9]+)"
         replacementPattern="#xpath(//div1[@n='$1']/div2[@n='$2']/div3[@n='$3']/l[@n='$4']">
        <p>A canonical reference is assembled with
        <list>
           <item>the name of the <label>work</label>: the
           <att>n</att> of a <gi>div1</gi>,</item>
           <item>a space,</item>
           <item>the number of the <label>book</label>: the
           <att>n</att> of a child <gi>div2</gi>,</item>
           <item>a full stop</item>
           <item>the number of the <label>poem</label>: the
```

```
          <att>n</att> of a child <gi>div3</gi>,</item>
          <item>the line number: the <att>n</att> value of a
                    child <gi>l</gi>
          </item>
        </list>
      </p>
    </cRefPattern>
    <cRefPattern
       matchPattern="([^ ]+) ([0-9]+)\.([0-9]+)"
       replacementPattern="#xpath(//div1[@n='$1']/div2[@n='$2']/div3[@n='$3']">
      <p>Same as above, but without the last component (full
            stop followed by the <gi>l</gi>'s <att>n</att>.</p>
    </cRefPattern>
    <cRefPattern
       matchPattern="([^ ]+) ([0-9]+)"
       replacementPattern="#xpath(//div1[@n='$1']/div2[@n='$2']">
      <p>Same as above, but without the poem component (full
            stop followed by the <gi>div3</gi>'s <att>n</att>.</p>
    </cRefPattern>
   </refsDecl>
  </encodingDesc>
</teiHeader>
```

The second example encodes the same reference system, again using the **n** attribute on the <div1>, <div2>, <div3>, and <l> tags, but giving the reference string in full on each tag. If canonical references are made only to lines, the reference system could be declared as follows:

```
<refsDecl>
  <cRefPattern
     matchPattern="([^ ]+ [0-9]+\.[0-9]+\.[0-9]+)"
     replacementPattern="#xpath(//l[@n='$1')"/>
</refsDecl>
```

Since the entire regular expression is enclosed as a parenthetical subgroup, the entire canonical reference string is sought as the value of the **n** attribute on an <l> element.

In order to handle references to poems as well as to individual lines, the declaration for the reference system must be more complicated:

```
<refsDecl>
  <cRefPattern
     matchPattern="([^ ]+ [0-9]+\.[0-9]+\.[0-9]+)"
     replacementPattern="#xpath(//l[@n='$1')"/>
  <cRefPattern
     matchPattern="([^ ]+ [0-9]+\.[0-9]+)"
     replacementPattern="#xpath(//div2[@n='$1')"/>
</refsDecl>
```

This declaration indicates that the entire reference string must be sought as the value of the **n** attribute on a <div1>, <div2>, <div3>, or <l> element.

The third example encodes the same reference system, this time giving the entire reference string as the value of the **xml:id** attribute on the relevant tags. The reference system declaration for such an encoding could be:

```
<refsDecl>
  <cRefPattern  matchPattern="(.*)"  replacementPattern="#$1"/>
</refsDecl>
```

although in general there seems to be little advantage in this case: it is no more difficult to use a standard relative URI reference as the value of **target**.

Reference systems recorded by means of milestone tags can also be declared; the following prose description could be used to declare the example given in section *3.10.3. Milestone Elements*.

```
<refsDecl>
  <p>Standard references to work, book, poem, and line may be
     constructed from the milestone tags in the text.</p>
</refsDecl>
```

Or in this way, using a formal declaration for this reference scheme derived from edition **E1**.

```
<refsDecl>
  <refState  ed="E1"  unit="work"  delim=" "/>
  <refState  ed="E1"  unit="book"  delim="."/>
  <refState  ed="E1"  unit="poem"  delim=":"/>
  <refState  ed="E1"  unit="line"/>
</refsDecl>
```

3.11 Bibliographic Citations and References

Bibliographic references (that is, full descriptions of bibliographic items such as books, articles, films, broadcasts, songs, etc.) or pointers to them may appear at various places in a TEI text. They are required at several points within the TEI Header's source description, as discussed in section *2.2.7. The Source Description*; they may also appear within the body of a text, either singly (for example within a footnote), or collected together in a list as a distinct part of a text; detailed bibliographic descriptions of manuscript or other source materials may also be required. These Guidelines propose a number of specialised elements to encode such descriptions, which together constitute the model.biblLike class. By default, this class has the following members:

<bibl> (bibliographic citation) contains a loosely-structured bibliographic citation of which the sub-components may or may not be explicitly tagged.

<biblStruct> (structured bibliographic citation) contains a structured bibliographic citation, in which only bibliographic sub-elements appear and in a specified order.

<biblFull> (fully-structured bibliographic citation) contains a fully-structured bibliographic citation, in which all components of the TEI file description are present.

Lists of such elements may also be encoded using the following element:

<listBibl> (citation list) contains a list of bibliographic citations of any kind.

In printed texts, the individual constituents of a bibliographic reference are conventionally marked off from each other and from the flow of text by such features as bracketing, italics, special punctuation conventions, underlining, etc. In electronic texts, such distinctions are also important, whether in order to produce acceptably formatted output or to facilitate intelligent retrieval processing,[6] quite apart from the need to distinguish the reference itself as a textual object with particular linguistic properties.

[6] For example, to distinguish *London* as an author's name from *London* as a place of publication or as a component of a title.

It should be emphasized that for references as for other textual features, the primary or sole consideration is not how the text should be formatted when it is printed. The distinctions permitted by the scheme outlined here may not necessarily be all that particular formatters or bibliographic styles require, although they should prove adequate to the needs of many such commonly used software systems.[7] The features distinguished and described below (in section *3.11.2. Components of Bibliographic References*) constitute a set which has been useful for a wide range of bibliographic purposes and in many applications, and which moreover corresponds to a great extent with existing bibliographic and library cataloguing practice. For a fuller account of that practice as applied to electronic texts see section *2.2.7. The Source Description*; for a brief mention of related library standards see section *2.7. Note for Library Cataloguers*.

3.11.1 Elements of Bibliographic References

The members of the model.biblLike class all share a number of possible component sub-elements. For the <bibl> and <biblStruct> elements, exactly the same sub-elements are concerned, and they are described together in section *3.11.2. Components of Bibliographic References*; for the <biblFull> element, the sub-elements concerned are fully described in section *2.2. The File Description*.

Different levels of specific tagging may be appropriate in different situations. In some cases, it may be felt necessary to mark just the extent of the reference itself, with perhaps a few distinctions being made within it (for example, between the part of the reference which identifies a title or author and the rest). Such references, containing a mixture of text with specialized bibliographic elements, are regarded as <bibl> elements, and tagged accordingly. For example:

```
<p>A book which had a great influence on him
was <bibl>Tufte's <title>Envisioning
        Information</title>
  </bibl>, although he may
never have actually read it.</p>
```

Indeed, some encoders may find it unnecessary to mark the bibliographic reference at all:

```
<p>A book which had a great influence on him
was Tufte's <title>Envisioning Information</title>,
although he may never have actually read it.</p>
```

Some bibliographic references are extremely elliptical, often only a string of the form *Baxter, 1983*. If no further details of Baxter's book are given in the source text and none are supplied by the encoder, then the reference thus given should be tagged as a <bibl>:

```
All of this is of course much more fully treated
in <bibl>Baxter, 1983</bibl>.
```

In general, however, normal modern bibliographic practice, and these Guidelines, distinguish between a bibliographic reference, which is a self-sufficient description of a bibliographic item, and a bibliographic pointer, which is a short-form citation (e.g. *Baxter, 1983*) which serves usually as a place-holder or pointer to a full long-form reference found elsewhere in the text. The usual encoding of short-form references such as

[7] Among the bibliographic software systems and subsystems consulted in the design of the <biblStruct> structure were BibTeX, Scribe, and ProCite. The distinctions made by all three may be preserved in <biblStruct> structures, though the nature of their design prevents a simple one-to-one mapping from their data elements to TEI elements. For further information, see section *3.11.4. Relationship to Other Bibliographic Schemes*.

Baxter, 1983 is not as <bibl> elements but as cross-references to such elements; see section *3.11.3. Bibliographic Pointers* below.

In cases where the encoder wishes to impose more structure on the bibliographic information, for example to make sure it conforms to a particular stylesheet or retrieval processor, the <biblStruct> element should be used. Note that several of the features in this and later examples are explained later in the current section.

```
<biblStruct>
  <monogr>
    <author>Edward R. Tufte</author>
    <title>Envisioning Information</title>
    <imprint>
      <pubPlace>Cheshire, Conn.</pubPlace>
      <publisher>Graphics Press</publisher>
      <date>1990</date>
    </imprint>
  </monogr>
</biblStruct>
```

Source: [199]

A more complex and detailed bibliographic structure is provided by the <biblFull> element defined in the TEI header module. This element is provided as a means of embedding the file description of one existing digital text within that of another (see further section *2.2. The File Description*); however, its use is not confined to digital texts, and it may be used in the same way as any other bibliographic element, as in this example:

```
<biblFull>
  <titleStmt>
    <title>Envisioning Information</title>
    <author>Tufte, Edward R[olf]</author>
  </titleStmt>
  <extent>126 pp.</extent>
  <publicationStmt>
    <publisher>Graphics Press</publisher>
    <pubPlace>Cheshire, Conn. USA</pubPlace>
    <date>1990</date>
  </publicationStmt>
</biblFull>
```

Source: [199]

A list of bibliographic items, of whatever kind, may be treated in the same way as any other list (see section *3.7. Lists*). Alternatively, the specialized <listBibl> element may be used. The difference between the two is that a <list> contains <item> elements, within which bibliographic elements (<bibl>, <biblStruct>, or <biblFull>) may appear, as well as other phrase- and paragraph-level elements, whereas the <listBibl> may contain only bibliographic elements, optionally preceded by a heading and a series of introductory paragraphs. The former would be appropriate for a list of bibliographic elements in which descriptive prose predominated, and the latter for a more formal bibliography. The following are thus both legal encodings of a list of bibliographic entries: a <listBibl>:

```
<listBibl>
  <head>Bibliography</head>
  <biblStruct  xml:id="NELSON80">
    <analytic>
```

```
     <author>Nelson, T. H.</author>
     <title>Replacing the printed word:
           a complete literary system.</title>
   </analytic>
   <monogr>
     <title>Information Processing '80: Proceedings of the IFIPS
           Congress, October 1980</title>
     <editor>Simon H. Lavington</editor>
     <imprint>
       <publisher>North-Holland</publisher>
       <pubPlace>Amsterdam</pubPlace>
       <date>1980</date>
     </imprint>
     <biblScope>pp 1013–23 </biblScope>
   </monogr>
   <note>Apparently a draft of section 4 of
   <title>Literary Machines</title>.</note>
 </biblStruct>
 <bibl  xml:id="NELSON88">Ted Nelson: <title>Literary Machines</title>
    (privately published, 1987)</bibl>
 <bibl  xml:id="BAXTER88">
   <author>Baxter, Glen</author>
   <title>Glen Baxter His Life: the years of struggle</title>
    London: Thames and Hudson, 1988.
 </bibl>
</listBibl>
```

or a simple <list>:

```
<list>
 <head>Bibliography</head>
 <item>
   <bibl  xml:id="NEL80">
     <author>Nelson, T. H.</author>
     <title  level="a">Replacing the printed word:
           a complete literary system.</title>
     <title  level="m">Information Processing '80:
           Proceedings of the IFIPS Congress, October 1980</title>
     <editor>Simon H. Lavington</editor>
     <publisher>North-Holland</publisher>
     <pubPlace>Amsterdam</pubPlace>
     <date>1980</date>
     <biblScope>pp 1013–23
     </biblScope>
     <note>Apparently a draft of section 4 of
     <title>Literary Machines</title>.</note>
   </bibl>
 </item>
 <item>
   <bibl  xml:id="NEL88">Ted Nelson: <title>Literary Machines</title>
       (privately published, 1987)</bibl>
 </item>
 <item>
   <bibl  xml:id="BAX88">
     <author>Baxter, Glen</author>
     <title>Glen Baxter His Life: the years of struggle</title>
```

```
        London: Thames and Hudson, 1988.
    </bibl>
  </item>
</list>
```

3.11.2 Components of Bibliographic References

This section discusses a number of very commonly occurring component elements of bibliographic references. They fall into four groups:

- elements for grouping components of the analytic, monographic, and series levels in a structured bibliographic reference
- titles of various kinds, and statements of intellectual responsibility (authorship, etc.)
- information relating to the publication, pagination, etc. of an item (most of these constitute the default members of the model.biblPart class)
- annotation, commentary, and further detail

The following sections describe the elements which may be used to represent such information within a <bibl> or <biblStruct> element. Within the former, elements from the model.biblPart class, other phrase-level elements, and plain text may be combined without other constraint; within the latter, such of these elements as exist for a given reference must be distinguished, and must also be presented in a specific order, discussed further below (section *3.11.2.7. Order of Components within References*).

3.11.2.1 Analytic, Monographic, and Series Levels

In common library practice a clear distinction is made between an individual item within a larger collection and a free-standing book, journal, or collection. Similarly a book in a series is distinguished sharply from the series within which it appears. An article forming part of a collection which itself appears in a series thus has a bibliographic description with three quite distinct levels of information:

1. the analytic level, giving the title, author, etc., of the article;
2. the monographic level, giving the title, editor, etc., of the collection;
3. the series level, giving the title of the series, possibly the names of its editors, etc., and the number of the volume within that series.

In the same way, an article in a journal requires at least two levels of information: the analytic level describing the article itself, and the monographic level describing the journal.

These three levels may be distinguished within a <bibl> element, and must be distinguished within a <biblStruct> element if present, by means of the following elements:

<analytic> (analytic level) contains bibliographic elements describing an item (e.g. an article or poem) published within a monograph or journal and not as an independent publication.

<monogr> (monographic level) contains bibliographic elements describing an item (e.g. a book or journal) published as an independent item (i.e. as a separate physical object).

<series> (series information) contains information about the series in which a book or other bibliographic item has appeared.

For purposes of TEI encoding, journals and anthologies are both treated as monographs; a journal title will thus be tagged as a **<title level="j">** element, or simply as a <title> within a <monogr> element. Individual articles in the journal or collected texts should be treated at the 'analytic' level. When an article has been printed in more than one journal or collection, the bibliographic reference may have more than one <monogr> element,

each possibly followed by one or more <series> elements. A <series> element always relates to the most recently preceding <monogr> element. (Whether reprints of an article are treated in the same bibliographic reference or a separate one varies among different styles. Library lists typically use a different entry for each publication, while academic footnoting practice typically treats all publications of the same article in a single entry.)

For example, the article cited in this example has been published twice, once in a journal and once in a collection which appeared in a German language series:

```
<biblStruct>
  <analytic>
    <author>Thaller, Manfred</author>
    <title  level="a">A Draft Proposal for a Standard for the
        Coding of Machine Readable Sources</title>
  </analytic>
  <monogr>
    <title  level="j">Historical Social Research</title>
    <imprint>
      <biblScope  type="vol">40</biblScope>
      <date>October 1986</date>
      <biblScope  type="pages">3-46</biblScope>
    </imprint>
  </monogr>
  <monogr>
    <title  level="m">Modelling Historical Data:
        Towards a Standard for Encoding and
        Exchanging Machine-Readable Texts</title>
    <editor>Daniel I. Greenstein</editor>
    <imprint>
      <pubPlace>St. Katharinen</pubPlace>
      <publisher>Max-Planck-Institut für Geschichte
            In Kommission bei
            Scripta Mercaturae Verlag</publisher>
      <date>1991</date>
    </imprint>
  </monogr>
  <series  xml:lang="de">
    <title  level="s">Halbgraue Reihe
        zur Historischen Fachinformatik</title>
    <respStmt>
      <resp>Herausgegeben von</resp>
      <name  type="person">Manfred Thaller</name>
      <name  type="org">Max-Planck-Institut für Geschichte</name>
    </respStmt>
    <title  level="s">Serie A: Historische Quellenkunden</title>
    <biblScope>Band 11</biblScope>
  </series>
</biblStruct>
```

The practice of analytic vs. monographic citation, as described here, should be distinguished from the practice of including within one citation a reference to another work, which the encoder considers to be related to in some way: see further *3.11.2.5. Related items* below.

Punctuation should not appear between the elements within a structured bibliographic entry, unless it is contained within the elements it delimits. As the example shows, it is possible to encode the entry without any inter-element punctuation: this facilitates use of the <biblStruct> element in systems which can render bibliographic references in any of several styles.

3.11.2.2 Authors, Titles, and Editors

Bibliographic references typically begin with a statement of the title being cited followed by the names of those intellectually responsible for it. For articles in journals or collections, such statements should appear both for the analytic and for the monographic level. The following elements are provided for tagging such elements:

<title> contains the full title of a work of any kind.

<author> in a bibliographic reference, contains the name of the author(s), personal or corporate, of a work; the primary statement of responsibility for any bibliographic item.

<editor> secondary statement of responsibility for a bibliographic item, for example the name of an individual, institution or organization, (or of several such) acting as editor, compiler, translator, etc.

<respStmt> (statement of responsibility) supplies a statement of responsibility for the intellectual content of a text, edition, recording, or series, where the specialized elements for authors, editors, etc. do not suffice or do not apply.

<resp> (responsibility) contains a phrase describing the nature of a person's intellectual responsibility.

<name> (name, proper noun) contains a proper noun or noun phrase.

<meeting> contains the formalized descriptive title for a meeting or conference, for use in a bibliographic description for an item derived from such a meeting, or as a heading or preamble to publications emanating from it.

The elements <author>, <editor>, and <respStmt> are the default members of the model.respLike class, a subclass of the model.biblPart class to which the constituents of the <bibl> element belong.

In bibliographic references, all titles should be tagged as such, whether analytic, monographic, or series titles. The single element <title> is used for all these cases. When it appears directly within an <analytic>, <monogr>, or <series> element, <title> is interpreted as belonging to the appropriate level. When it appears elsewhere, its **level** attribute should be used to signal its bibliographic level. It is a semantic error to give a value for the **level** attribute which is inconsistent with the context; such values may be ignored. The **level** value **a** implies the analytic level; the values **m**, **j**, and **u** imply the monographic level; the value **s** implies the series level. Note, however, that the semantic error occurs only if the nested title is directly enclosed by the <analytic>, <monogr>, or <series> element; if it is enclosed only indirectly (i.e., nested more deeply), no semantic error need be present. For example, the analytic title may contain a monographic title:

```
<biblStruct>
  <analytic>
    <author>Lucy Allen Paton</author>
    <title>Notes on Manuscripts of the
    <title  level="m"  xml:lang="fr">Prophécies de Merlin</title>
    </title>
  </analytic>
  <monogr>
    <title  level="j">PMLA</title>
    <imprint>
      <biblScope  type="vol">8</biblScope>
      <date>1913</date>
      <biblScope  type="pages">122</biblScope>
    </imprint>
  </monogr>
</biblStruct>
```

In this case, the analytic title 'Notes on Manuscripts of the *Prophécies de Merlin*' needs no **level** attribute because it is directly contained by the <analytic> level; the monographic title contained within it, 'Prophécies de Merlin,' does not create a semantic error because it is not directly contained by the <analytic> element.

In some bibliographic applications, it may prove useful to distinguish main titles from subordinate titles, parallel titles, etc. The **type** attribute is provided to allow this distinction to be recorded.

The following reference, from a national standard for bibliographic references, illustrates this type of analysis with its distinction between main and subordinate titles. Note that this uses the more flexible <bibl>, rather than the structured <biblStruct> element: consequently, there is no requirement to tag all the components of the reference (notably the authors).

```
<bibl>Saarikoski, Pirkko-Liisa, and Paavo Suomalainen,
<title level="a" type="main">Studies on the physiology of
    the hibernating hedgehog, 15</title>
 <title level="a" type="subordinate">Effects of seasonal
    and temperature changes on the in vitro glycerol release from
    brown adipose tissue</title>
 <title level="j">Ann. Acad. Sci. Fenn., Ser. A4</title>
 <date>1972</date>
 <biblScope type="vol">187</biblScope>
 <biblScope type="pp">1-4</biblScope>
</bibl>
```

Source: [4]

Slightly more complex is the distinction made below among main, subordinate, and parallel titles, in an example from the same source (p. 63). The punctuation and the bibliographic analysis are those given in ANSI Z39.29-1977; the punctuation is in the style prescribed by the International Standard Bibliographic Description (ISBD).[8] Again, it is only because this example uses <bibl> rather than <biblStruct>, that specific punctuation may be included between the component elements of the reference.

```
<bibl>Tchaikovsky, Peter Ilich.
<title level="m" type="main">The swan lake ballet</title>
= <title level="m" type="parallel" xml:lang="fr">Le lac des cygnes</title>
: <title level="m" type="subordinate" xml:lang="fr">grand ballet en 4 actes</title>
: <title level="m" type="subordinate">op. 20</title>
[Score].
New York: Broude Brothers; [1951] (B.B. 59). vi, 685 p.</bibl>
```

Source: [4]

The elements <author> and <editor> have, for printed books and articles, a fairly obvious significance; for other kinds of bibliographic items their proper usage may be less obvious. The <author> element should be used for the person or agency with primary responsibility for a work's intellectual content, and the element <editor> for an editor of the work. Thus an organization such as a radio or television station is usually accounted 'author' of a broadcast, for example, while the author of a Government report will usually be the agency which produced it.

For anyone else with responsibility for the work, the <respStmt> element should be used. The nature of the responsibility is indicated by means of a <resp> element, and the person, organization, etc. responsible by a <name>, <persName>, or <orgName> element. Strings such as 'unknown' may be encoded using the <rs> element.

[8] The analysis is not wholly unproblematic: as the text of the standard points out, the first subordinate title is subordinate only to the parallel title in French, while the second is subordinate to both the English main title and the French parallel title, without this relationship being made clear, either in the markup given in the example or in the reference structure offered by the standard.

At least one of the four naming elements (<name>, <persName>, <orgName>, or <rs>) and one <resp> element should be given within the <respStmt> element, followed optionally by any number of any of them.

Examples of secondary responsibility of this kind include the roles of illustrator, translator, encoder, and annotator. The <respStmt> element may also be used for editors, if it is desired to record the specific terms in which their role is described.

Examples of <author> and <editor> may be found in sections *3.11.1. Elements of Bibliographic References*, and *3.11.2.1. Analytic, Monographic, and Series Levels*; wherever <author> and <editor> may occur, the <respStmt> element may also occur. When one of these elements precedes or immediately follows a title, it applies to that title; when it follows an <edition> element or occurs within an edition statement, it applies to the edition in question.

In this example, the <respStmt> elements apply to the work as a whole, not merely to the first edition:

```
<bibl>
  <author>Lominadze, D. G.</author>
  <title  level="m">Cyclotron waves in plasma.</title>
  <respStmt>
    <resp>translated by</resp>
    <name>A. N. Dellis;</name>
  </respStmt>
  <respStmt>
    <resp>edited by</resp>
    <name>S. M. Hamberger.</name>
  </respStmt>
  <edition>1st ed.</edition>
  <pubPlace>Oxford:</pubPlace>
  <publisher>Pergamon Press,</publisher>
  <date>1981.</date>
  <extent>206 p.</extent>
  <title  level="s">International series in natural philosophy.</title>
  <note  place="inline">Translation of:
  <title  xml:lang="ru"  level="m">Ciklotronnye volny v plazme.</title>
  </note>
</bibl>
```

Source: [104]

In this example, by contrast, the <respStmt> element applies to the edition, and not to the collection per se (Moser and Tervooren were not responsible for the first thirty-five printings); the elements of the reference have been reordered from their appearance on the title page of the volume in order to ensure the correct relationship of the collection title, the edition statement, and the statement of responsibility.

```
<biblStruct>
  <monogr  xml:lang="de">
    <title>Des Minnesangs Frühling</title>
    <note  place="inline">Mit 1 Faksimile</note>
    <edition>36., neugestaltete und erweiterte Auflage</edition>
    <respStmt>
      <resp>Unter Benutzung der Ausgaben von <name>Karl
               Lachmann</name> und <name>Moriz Haupt</name>, <name>Friedrich
               Vogt</name> und <name>Carl von Kraus</name> bearbeitet von</resp>
      <name>Hugo Moser</name>
      <name>Helmut Tervooren</name>
    </respStmt>
    <imprint>
```

```
    <biblScope  type="volume">I Texte</biblScope>
    <pubPlace>Stuttgart</pubPlace>
    <publisher>S. Hirzel Verlag</publisher>
    <date>1977</date>
   </imprint>
  </monogr>
</biblStruct>
```

Another form of 'responsibility' arises when a work is published as the outcome of a conference, workshop or similar meeting. The <meeting> element may be used to supply this information, as in the following example:

```
<biblStruct>
  <monogr>
   <title>Proceedings of a workshop on corpus resources</title>
   <respStmt>
    <resp>Programme Organizer</resp>
    <name>Geoffrey Leech</name>
   </respStmt>
   <meeting>DTI Speech and Language Technology Club meeting, 3-4
       January 1990, Wadham College, Oxford</meeting>
  </monogr>
</biblStruct>
```

3.11.2.3 Imprint, Pagination, and Other Details

By *imprint* is meant all the information relating to the publication of a work: the person or organization by whose authority and in whose name a bibliographic entity such as a book is made public or distributed (whether a commercial publisher or some other organization), the place of publication, and a date. It may also include a full address for the publisher or organization. Full bibliographic references usually specify either the number of pages in a print publication (or equivalent information for non-print materials), or the specific location of the material being cited within its containing publication. The following elements are provided to hold this information:

<imprint> groups information relating to the publication or distribution of a bibliographic item.

<address> contains a postal address, for example of a publisher, an organization, or an individual.

<pubPlace> (publication place) contains the name of the place where a bibliographic item was published.

<publisher> provides the name of the organization responsible for the publication or distribution of a bibliographic item.

<date> contains a date in any format.

<idno> (identifying number) supplies any standard or non-standard number used to identify a bibliographic item.

<extent> describes the approximate size of a text as stored on some carrier medium, whether digital or non-digital, specified in any convenient units.

<biblScope> (scope of citation) defines the scope of a bibliographic reference, for example as a list of page numbers, or a named subdivision of a larger work.

The elements <biblScope>, <pubPlace> and <publisher> constitute the special class model.imprintPart; members of this class may appear with a date inside an <imprint> element in a specific location within a <biblStruct>, or alternatively, they may appear alongside any other bibliographic component inside a <bibl>.

For bibliographic purposes, usually only the place (or places) of publication are required, possibly including the name of the country, rather than a full address; the element <pubPlace> is provided for this purpose. Where however the full postal address is likely to be of importance in identifying or locating the bibliographic item concerned, it may be supplied and tagged using the <address> element described in section *3.5.2. Addresses*. Alternatively, if desired, the <rs> or <name> elements described in section *3.5.1. Referring Strings* may be used; this involves no claim that the information given is either a full address or the name of a city.

The name of the publisher of an item should be marked using the <publisher> element even if the item is made public ('published') by an organization other than a conventional publisher, as is frequently the case with technical reports:

```
<biblStruct>
  <monogr>
    <author>Nicholas, Charles K.</author>
    <author>Welsch, Lawrence A.</author>
    <title>On the interchangeability of SGML and ODA</title>
    <imprint>
      <pubPlace>Gaithersburg, MD</pubPlace>
      <publisher>National Institute of Standards and Technology
      </publisher>
      <date  when="1992-01">January 1992</date>
    </imprint>
    <extent>19 pp.</extent>
  </monogr>
  <idno  type="NIST">NISTIR 4681</idno>
</biblStruct>
```

and with dissertations:

```
<biblStruct>
  <monogr>
    <author>Hansen, W.</author>
    <title  level="u">Creation of hierarchic text
        with a computer display</title>
    <note  place="inline">Ph.D. dissertation</note>
    <imprint>
      <publisher>Dept. of Computer Science, Stanford Univ.</publisher>
      <pubPlace>Stanford, CA</pubPlace>
      <date  when="1971-06">June 1971</date>
    </imprint>
  </monogr>
</biblStruct>
```

Source: [163]

When an item has been reprinted, especially reprinted without change from a specific earlier edition, the reprint may appear in a <monogr> element with only the <imprint> and other details of the reprint. In the following example, a microform reprint has been issued without any change in the title or authorship. The series statement here applies only to the second <monogr> element.

```
<biblStruct>
  <monogr>
    <author>Shirley, James</author>
    <title  type="main">The gentlemen of Venice</title>
```

```
    <title type="subordinate">a tragi-comedie presented at the private
        house in Salisbury Court by Her Majesties servants</title>
    <note place="inline">[Microform]</note>
    <imprint>
      <pubPlace>London</pubPlace>
      <publisher>H. Moseley</publisher>
      <date>1655</date>
    </imprint>
    <extent>78 p.</extent>
  </monogr>
  <monogr>
    <imprint>
      <pubPlace>New York</pubPlace>
      <publisher>Readex Microprint</publisher>
      <date>1953</date>
    </imprint>
    <extent>1 microprint card, 23 x 15 cm.</extent>
  </monogr>
  <series>
    <title>Three centuries of drama: English, 1642–1700</title>
  </series>
</biblStruct>
```

Source: [4]

An alternative way of handling the above situation would be to use the <relatedItem> element described in section *3.11.2.5. Related items* below.

A bibliographic description, particularly for an analytic title, will often include some additional information specifying its location, for example as a volume number, page number, range of page numbers, or name or number of a subdivision of the host work. The element <biblScope> may be used to identify such information if it is present. Where it is desired to distinguish different classes of such information (volume number, page number, chapter number, etc.), the **type** attribute may be used with any convenient typology.

When the item being cited is a journal article, the <imprint> element describing the issue in which it appeared may contain <biblScope> elements for volume and page numbers, together with a <date> element.

For example:

```
<biblStruct>
  <analytic>
    <author>Wrigley, E. A.</author>
    <title>Parish registers and the historian</title>
  </analytic>
  <monogr>
    <editor>Steel, D. J.</editor>
    <title>National index of parish registers</title>
    <imprint>
      <pubPlace>London</pubPlace>
      <publisher>Society of Genealogists</publisher>
      <date when="1968">1968</date>
      <biblScope type="vol">vol. 1</biblScope>
      <biblScope type="pp">pp. 155–167.</biblScope>
    </imprint>
  </monogr>
</biblStruct>
```

The **type** attribute on <biblScope> is optional: both the following are legal examples:

```
<biblStruct>
  <analytic>
    <author>Boguraev, Branimir</author>
    <author>Neff, Mary</author>
    <title>Text Representation, Dictionary Structure,
        and Lexical Knowledge</title>
  </analytic>
  <monogr>
    <title  level="j">Literary & Linguistic Computing</title>
    <imprint>
      <biblScope  type="vol">7</biblScope>
      <biblScope  type="issue">2</biblScope>
      <date>1992</date>
      <biblScope  type="pp">110-112</biblScope>
    </imprint>
  </monogr>
</biblStruct>
```

```
<biblStruct>
  <analytic>
    <author>Chesnutt, David</author>
    <title>Historical Editions in the States</title>
  </analytic>
  <monogr>
    <title  level="j">Computers and the Humanities</title>
    <imprint>
      <biblScope>25.6</biblScope>
      <date  when="1991-12">(December, 1991):</date>
      <biblScope>377–380</biblScope>
    </imprint>
  </monogr>
</biblStruct>
```

3.11.2.4 Series Information

Series information may (in <bibl> elements) or must (in <biblStruct> elements) be enclosed in a <series> element or (in a <biblFull> element) a <seriesStmt> element. The title of the series may be tagged **<title level="s">**, the volume number **<biblScope type="vol">**, and responsibility statements for the series (e.g. the name and affiliation of the editor, as in the example in section *3.11.2.1. Analytic, Monographic, and Series Levels*) may be tagged <editor> or <respStmt>.

3.11.2.5 Related items

In bibliographic parlance, a related item is any bibliographic item which, though related to that being defined, is distinct from it. The distinction between analytic and monographic items made above may be thought of as a special case of this kind of 'related' item. More usually however, the term is applied to such items as translations, continuations, original sources, parts, etc.

The element <relatedItem> is provided as a means of documenting such associated items:

<relatedItem> contains or references some other bibliographic item which is related to the present one in some specified manner, for example as a constituent or alternative version of it.

In the following example, the first <biblStruct> describes a facsimile edition, and the second describes the work of which it is a facsimile. The relation between the facsimile and its source is represented by means of a <relatedItem> within the first description, which points to the description of the source.

```
<biblStruct xml:id="bibl03">
  <monogr>
    <author>Swinburne, Algernon Charles</author>
    <title>Swinburne's <title>Atalanta in Calydon</title>: A Facsimile of the
        First Edition</title>
    <editor>Georges Lafourcade</editor>
    <imprint>
      <pubPlace>London</pubPlace>
      <publisher>Oxford UP</publisher>
      <date>1930</date>
    </imprint>
  </monogr>
  <relatedItem type="original">
    <ref target="#bibl04"/>
  </relatedItem>
</biblStruct>
<biblStruct xml:id="bibl04">
  <monogr>
    <author> Swinburne, Algernon Charles</author>
    <title>Atalanta in Calydon</title>
    <imprint>
      <pubPlace>London</pubPlace>
      <publisher>Edward Moxon</publisher>
      <date>1865</date>
    </imprint>
  </monogr>
</biblStruct>
```

The <ref> element in the above example could be replaced by the referenced <biblStruct> itself since a <relatedItem> may contain any form of bibliographic reference. For example, one of the examples quoted above might also be encoded as follows:

```
<biblStruct>
  <monogr>
    <author>Shirley, James</author>
    <title type="main">The gentlemen of Venice</title>
    <imprint>
      <pubPlace>New York</pubPlace>
      <publisher>Readex Microprint</publisher>
      <date>1953</date>
    </imprint>
    <extent>1 microprint card, 23 x 15 cm.</extent>
  </monogr>
  <series>
    <title>Three centuries of drama: English, 1642–1700</title>
  </series>
  <relatedItem type="original">
    <biblStruct>
      <monogr>
        <author>Shirley, James</author>
        <title type="main">The gentlemen of Venice</title>
        <title type="subordinate">a tragi-comedie presented at the private
               house in Salisbury Court by Her Majesties servants</title>
        <imprint>
          <pubPlace>London</pubPlace>
          <publisher>H. Moseley</publisher>
```

```
        <date>1655</date>
       </imprint>
       <extent>78 p.</extent>
     </monogr>
   </biblStruct>
  </relatedItem>
</biblStruct>
```

3.11.2.6 Notes and Other Additional Information

Explanatory notes about the publication of unusual items, the form of an item (e.g. *[Score]* or *[Microform]*), or its provenance (e.g. *translation of ...*) may be tagged using the <note> element. The same element may be used for any descriptive annotation of a bibliographic entry in a database.

<note> contains a note or annotation.

For example:

```
<bibl>
 <author>Coombs, James H., Allen H. Renear,
   and Steven J. DeRose.</author>
 <title  level="a">Markup Systems and the Future of Scholarly
   Text Processing.</title>
 <title  level="j">Communications of the ACM</title>
 <biblScope>30.11 (November 1987): 933–947.</biblScope>
 <note>Classic polemic supporting descriptive over procedural
   markup in scholarly work.</note>
</bibl>
```

3.11.2.7 Order of Components within References

The order of elements in <bibl> elements is not constrained.

In <biblStruct> elements, the <analytic> element, if it occurs, must come first, followed by one or more <monogr> and <series> elements, which may appear intermingled (as long as a <monogr> element comes first). Within <analytic>, the title(s), author(s), editor(s), and other statements of responsibility may appear in any order; it is recommended that all forms of the title be given together. Within <monogr>, the author, editor, and statements of responsibility may either come first or else follow the monographic title(s). Following these, the elements must appear in the following order:

- <note>s on the publication (and <meeting> elements describing the conference, in the case of a proceedings volume)
- <edition> elements, each followed by any related <editor> or <respStmt> elements
- <imprint>
- <biblScope>

Within <imprint>, the elements allowed may appear in any order.

Finally, within the <series> information in a <biblStruct>, the sequence of elements is not constrained.

If more detailed structuring of a bibliographic description is required, the <biblFull> element should be used. This is not further described here, as its contents are essentially equivalent to those of the <fileDesc> element in the <teiHeader>, which is fully described in section *2.2. The File Description*.

3.11.3 Bibliographic Pointers

References which are pointers to bibliographic items, of whatever kind, should be treated in the same way as other cross-references (see section *3.6. Simple Links and Cross-References*). As discussed in that section, cross-

referencing within TEI texts is in general represented by means of <ptr> or <ref> elements. A **target** attribute on these elements is used to supply an identifying value for the target of the cross-reference, which should be, in the case of bibliographic elements, a bibliographic reference of some kind. Where the form of the reference itself is unimportant, or may be reconstructed mechanically, or is not to be encoded, the <ptr> element is used, as in the following example:

```
As shown above (<ptr  target="#NEL80"/>) ...
```

Where the form of the reference is important, or contains additional qualifying information which is to be kept but distinguished from the surrounding text, the <ref> element should be used, as in the following example:

```
Nelson claims <ref  target="#NEL80">(ibid, passim)</ref> ...
```

It may be important to distinguish between the short form of a bibliographic reference and some qualifying or additional information. The latter should not appear within the scope of the <ref> element when this is the case, as for example in an application concerned to normalize bibliographic references:

```
Nelson claims (<ref  target="#NEL80">Nelson [1980]</ref> pages 13–37) ...
```

3.11.4 Relationship to Other Bibliographic Schemes

The bibliographic tagging defined here can capture the distinctions required by most bibliographic encoding systems; for the benefit of users of some commonly used systems, the following lists of equivalences are offered, showing the relationship of the markup defined here to the fields defined for bibliographic records in the Scribe, BibTeX, and ProCite systems.

Listed below are the equivalences between the various bibliographic fields defined for use in the Scribe and BibTeX systems of bibliographic databases and the elements defined in this module.[9] Elements and structures available in the module defined here which have no analogues in Scribe and BibTeX are not noted.

address tag as <placeName> or <address>

annote tag as <note>

author tag as <author>

booktitle tag as **<title level="m">** or <title> within <monogr>

chapter tag as **<biblScope type="chapter">**

date used only to record date entry was made in the bibliographic database; not supported

edition tag as <edition>

editor tag as <editor> or <respStmt>

editors tag as multiple <editor> or <respStmt> elements

[9] The BibTeX scheme is intentionally compatible with that of Scribe, although it omits some fields used by Scribe. Hence only one list of fields is given here.

fullauthor use the <reg> element, possibly inside a <choice> element, inside either an <author> or <name>

fullorganization use the <reg> element, possibly inside a <choice> element, inside a **<name type="org">**

howpublished tag as <note>, possibly using the form **<note place="inline">**

institution used only for issuer of technical reports; tag as <publisher>

journal tag as **<title level="j">** or <title> within <monogr>

key used to specify an alternate sort key for the bibliographic item, for use instead of author's or editor's name; not supported

meeting tag as <meeting> or as <note>

month use <date>; if the date is not in a trivially parseable form, use the **when** attribute to provide a normalized equivalent in one of the format from *XML Schema Part 2: Datatypes Second Edition*

note tag as <note>

number tag as **<biblScope type="issue">** or **<biblScope type="number">**; for technical report numbers, use **<idno type="docno">**

organization used only for sponsor of conference; use **<name type="org">** within <respStmt> within <meeting> element

pages tag as **<biblScope type="pp">**

publisher tag as <publisher>

school used only for institutions at which thesis work is done; tag as <publisher>

series tag as **<title level="s">** or <title> within <series>

title tag as <title> in appropriate context or with appropriate **level** value

volume tag as **<biblScope type="vol">**

year tag as <date>; if the date is not in a trivially parseable form, use the **when** attribute to provide an ISO-format equivalent

3.12 Passages of Verse or Drama

The following elements are included in the core module for the convenience of those encoding texts which include mixtures of prose, verse and drama.

<l> (verse line) contains a single, possibly incomplete, line of verse.

<lg> (line group) contains a group of verse lines functioning as a formal unit, e.g. a stanza, refrain, verse paragraph, etc.

<sp> (speech) An individual speech in a performance text, or a passage presented as such in a prose or verse text.

<speaker> A specialized form of heading or label, giving the name of one or more speakers in a dramatic text or fragment.

<stage> (stage direction) contains any kind of stage direction within a dramatic text or fragment.

Full details of other, more specialized, elements for the encoding of texts which are predominantly verse or drama are described in the appropriate chapter of part three (for verse, see the verse base described in chapter *6. Verse*; for performance texts, see the drama base described in chapter *7. Performance Texts*). In this section, we describe only the elements listed above, all of which can appear in any text, whichever of the three modes prose, verse, or drama may predominate in it.

3.12.1 Core Tags for Verse

Like other written texts, verse texts or poems may be hierarchically subdivided, for example into books or cantos. These structural subdivisions should be encoded using the general purpose or <div1> (etc.) elements described below in chapters *4. Default Text Structure* and *6. Verse*. The fundamental unit of a verse text is the verse line rather than the paragraph, however.

The <l> element is used to mark up verse lines, that is metrical rather than typographic lines. In some modern or free verse, it may be hard to decide whether the typographic line is to be regarded as a verse line or not, but the distinction is quite clear for verse following regular metrical patterns. Where a metrical line is interrupted by a typographic line break, the encoder may choose to ignore the fact entirely or to use the empty <lb> (line break) element discussed in *3.10. Reference Systems*. By convention, the start of a metrical line implies the start of a typographic line; hence there is no need to introduce an <lb> tag at the start of every <l> element, but only at places where a new typographic line starts within a metrical line, as in the following example:

```
<l>Of Mans First Disobedience, and<lb/> the Fruit</l>
<l>Of that Forbidden Tree, whose<lb/> mortal tast</l>
<l>Brought Death into the World,<lb/> and all our woe,</l>
<l>With loss of Eden, till one greater Man</l>
<l>Restore us, and regain the blissful Seat...</l>
```

Source: [143]

In the original copy text, the presence of an ornamental capital at the start of the poem means that the measure is not wide enough to print the first four lines on four lines; instead each metrical line occupies two typographic lines, with a break at the point indicated. Note that this encoding makes no attempt to preserve information about the whitespace or indentation associated with either kind of line; if regarded as essential, this information would be recorded using the **rend** or **rendition** attributes discussed in *1.3.1.1. Global Attributes*.

The <l> element should not be used to represent typographic lines in non-verse materials: if the line-breaking points in a prose text are considered important for analysis, they should be marked with the <lb> element. Alternatively, a neutral segmentation element such as <seg> or <ab> may be used; see further discussion of these elements in chapter *16. Linking, Segmentation, and Alignment*. The <l> element is a member of the model.lLike class, which is a subclass of the model.divPart class, along with elements from the model.pLike (paragraph-like) class.

In some verse forms, regular groupings of lines are regarded as units of some kind, often identified by a regular verse scheme. In stichic verse and couplets, groups of lines analogous to paragraphs are often indicated by indentation. In other verse forms, lines are grouped into irregular sequences indicated simply by whitespace. The <lg> or line group element may be used to mark any such grouping of elements from the model.lLike class. As a member of the att.typed class, the <lg> element bears the following attributes:

att.typed provides attributes which can be used to classify or subclassify elements in any way.
- **@type** characterizes the element in some sense, using any convenient classification scheme or typology.
- **@subtype** provides a sub-categorization of the element, if needed

which may be used to further categorize the line group where this is felt desirable, as in the following example. This example also demonstrates the **rend** attribute to indicate whether or not a line is indented.

```
<lg>
  <l>Come fill up the Glass,</l>
  <l  rend="indent">Round, round let it pass,</l>
  <l>'Till our Reason be lost in our Wine:</l>
  <l  rend="indent">Leave Conscience's Rules</l>
  <l  rend="indent">To Women and Fools,</l>
  <l>This only can make us divine.</l>
</lg>
<lg  n="Chorus"  type="refrain">
  <l>Then a Mohock, a Mohock I'll be,</l>
  <l>No Laws shall restrain</l>
  <l>Our Libertine Reign,</l>
  <l>We'll riot, drink on, and be free.</l>
</lg>
```

Source: [80]

For some kinds of analysis, it may be useful to identify different kinds of line group within the same piece of verse. Such line groups may self-nest, in much the same way as the un-numbered element described in chapter *4. Default Text Structure*. For example:

```
<lg  type="sonnet">
  <lg  type="octet">
    <l>Thus speaks the Muse, and bends her brow severe:—</l>
    <l>“Did I, <name>Lætitia</name>, lend my choicest lays,</l>
    <l>And crown thy youthful head with freshest bays,</l>
    <l>That all the' expectance of thy full-grown year</l>
    <l>Should lie inert and fruitless? O revere</l>
    <l>Those sacred gifts whose meed is deathless praise,</l>
    <l>Whose potent charms the' enraptured soul can raise</l>
    <l>Far from the vapours of this earthly sphere!</l>
  </lg>
  <lg  type="sestet">
    <l>Seize, seize the lyre!  resume the lofty strain!</l>
    <l>'T is time, 't is time!  hark how the nations round</l>
    <l>With jocund notes of liberty resound,—</l>
    <l>And thy own <name>Corsica</name> has burst her chain!</l>
    <l>O let the song to <name>Britain's</name> shores rebound,</l>
    <l  rend="indent(-1)">Where Freedom's once-loved voice is heard,
        alas!  in vain.”</l>
  </lg>
</lg>
```

Source: [8]

It is often the case that verse line boundaries conflict with the boundaries of other structural elements. In the following example, the single verse line ‘A Workeman in't... welcome’ is interrupted by a stage direction:

```
<l>Thou fumblest <name>Eros</name>, and my Queenes a Squire</l>
<l>More tight at this, then thou: Dispatch.  O Loue,</l>
<l>That thou couldst see my Warres to day, and knew'st</l>
<l>The Royall Occupation, thou should'st see</l>
<l  part="I">A Workeman in't. <stage>Enter an Armed Soldier.</stage>
</l>
<l  part="F">Good morrow to thee, welcome.  </l>
```

Source: [171]

In this encoding, the **part** attribute is used, as with , to indicate that the last two <l> elements should be regarded as the initial and final parts of a single line, rather than as two lines.

The same technique may be used where verse lines are collected together into units such as verse paragraphs:

```
<lg  n="6"  type="para">
<!-- ... -->
  <l>Unprofitably travelling toward the grave,</l>
  <l>Like a false steward who hath much received</l>
  <l  part="I">And renders nothing back.</l>
</lg>
<lg  type="para"  n="7">
  <l  part="F">Was it for this</l>
  <l>That one, the fairest of all rivers, loved</l>
  <l>To blend his murmurs with my nurse's song,</l>
<!-- ... -->
</lg>
```

Source: [211]

The **part** attribute may also be attached to an <lg> element to indicate that it is incomplete, for example because it forms part of a group that is divided between two speakers, as in the following example:

```
<sp>
  <speaker>First Voice</speaker>
  <lg  type="stanza"  part="I">
    <l>But why drives on that ship so fast</l>
    <l>Withouten wave or wind?</l>
  </lg>
</sp>
<sp>
  <speaker>Second Voice</speaker>
  <lg  type="stanza"  part="F">
    <l>The air is cut away before,</l>
    <l>And closes from behind.</l>
  </lg>
</sp>
```

Source: [42]

For alternative methods of aligning groups of lines which do not form simple hierarchic groups, or which are discontinuous, see the more detailed discussion in chapter *16. Linking, Segmentation, and Alignment*. For discussion of other elements and attributes specific to the encoding of verse, see chapter *6. Verse*.

3.12.2 Core Tags for Drama

Like other written texts, dramatic and other performance texts such as cinema or TV scripts are often hierarchically organized, for example into acts and scenes. These structural subdivisions should be encoded using the general purpose or <div1> (etc.) elements described below in chapters *4. Default Text Structure* and *7. Performance Texts*. Within these divisions, the body of a performance text typically consists of speeches, often prefixed by a phrase indicating who is speaking, and occasionally interspersed with stage directions of various kinds.

In the following simple example, each speech consists of a single paragraph:

```
<div2  n="I.2"  type="scene">
  <head>Scene 2.</head>
  <stage  type="setting">Peachum, Filch.</stage>
  <sp>
    <speaker>FILCH.</speaker>
    <p>Sir, Black Moll hath sent word her Trial comes on in
        the Afternoon, and she hopes you will order Matters
        so as to bring her off.</p>
  </sp>
  <sp>
    <speaker>PEACHUM.</speaker>
    <p>Why, she may plead her Belly at worst; to my
        Knowledge she hath taken care of that Security.
        But, as the Wench is very active and industrious,
        you may satisfy her that I'll soften the Evidence.</p>
  </sp>
  <sp>
    <speaker>FILCH.</speaker>
    <p>Tom Gagg, sir, is found guilty.</p>
  </sp>
</div2>
```

Source: [80]

In the following example, each speech consists of a sequence of verse lines, some of them being marked as metrically incomplete:

```
<div1  n="I"  type="Act">
  <head>ACT I</head>
  <div2  n="1"  type="Scene">
    <head>SCENE I</head>
    <stage  rend="italic">Enter Barnardo and Francisco,
        two Sentinels, at several doors</stage>
    <sp>
      <speaker>Barn</speaker>
      <l  part="Y">Who's there?</l>
    </sp>
    <sp>
      <speaker>Fran</speaker>
      <l>Nay, answer me.  Stand and unfold yourself.</l>
    </sp>
    <sp>
      <speaker>Barn</speaker>
      <l  part="I">Long live the King!</l>
    </sp>
    <sp>
      <speaker>Fran</speaker>
      <l  part="M">Barnardo?</l>
    </sp>
    <sp>
      <speaker>Barn</speaker>
      <l  part="F">He.</l>
    </sp>
    <sp>
      <speaker>Fran</speaker>
      <l>You come most carefully upon your hour.</l>
```

```
   </sp>
   <sp>
     <speaker>Barn</speaker>
     <l>'Tis now struck twelve. Get thee to bed, Francisco.</l>
   </sp>
   <sp>
     <speaker>Fran</speaker>
     <l>For this relief much thanks. 'Tis bitter cold,</l>
     <l  part="I">And I am sick at heart.</l>
   </sp>
  </div2>
</div1>
```

Source: [175]

In some cases, as here in the First Quarto of *Hamlet*, the printed speaker attributions need to be supplemented by use of the **who** attribute; again, the lines are marked as complete or incomplete:

```
<stage>Enter two Centinels.
<add  place="right">Now call'd <name  xml:id="barnardo">Bernardo</name> &
  <name  xml:id="francisco">Francesco</name>.</add>
</stage>
<sp  who="#francisco">
  <speaker>1.</speaker>
  <l  part="Y">Stand: who is that?</l>
</sp>
<sp  who="#barnardo">
  <speaker>2.</speaker>
  <l  part="Y">Tis I.</l>
</sp>
<sp  who="#francisco">
  <speaker>1.</speaker>
  <l>O you come most carefully vpon your watch,</l>
</sp>
<sp  who="#barnardo">
  <speaker>2.</speaker>
  <l>And if you meete Marcellus and Horatio,</l>
  <l>The partners of my watch, bid them make haste.</l>
</sp>
<sp  who="#francisco">
  <speaker>1.</speaker>
  <l  part="Y">I will: See who goes there.</l>
</sp>
<stage>Enter Horatio and Marcellus.</stage>
```

Source: [176]

By contrast with the preceding examples, the following encodes an early printed edition without making any assumption about which parts are prose or verse:

```
<div1  n="I"  type="act">
  <div2  n="1"  type="scene">
    <head  rend="italic">Actus primus, Scena prima.</head>
    <stage  rend="italic"  type="setting">A tempestuous
        noise of Thunder and Lightning heard: Enter
```

```
        a Ship-master, and a Boteswaine.</stage>
    <sp>
      <speaker>Master.</speaker>
      <p>Bote-swaine.</p>
    </sp>
    <sp>
      <speaker>Botes.</speaker>
      <p>Heere Master: What cheere?</p>
    </sp>
    <sp>
      <speaker>Mast.</speaker>
      <p>Good: Speake to th' Mariners: fall
            too't, yarely, or we run our selues a ground,
            bestirre, bestirre. <stage  type="move">Exit.</stage>
      </p>
    </sp>
    <stage  type="move">Enter Mariners.</stage>
    <sp>
      <speaker>Botes.</speaker>
      <p>Heigh my hearts, cheerely, cheerely my harts: yare,
            yare: Take in the toppe-sale: Tend to th' Masters whistle:
            Blow till thou burst thy winde, if roome e-nough.</p>
    </sp>
  </div2>
</div1>
```

Source: [177]

The <sp> and <stage> elements should also be used to mark parts of a text otherwise in prose which are presented as if they were dialogue in a play. The following example is taken from a 19th century novel in which passages of narrative and passages of dialogue are mixed within the same chapter:

```
<sp>
  <speaker>The reverend Doctor Opimiam</speaker>
  <p>I do not think I have named a single unpresentable fish.</p>
</sp>
<sp>
  <speaker>Mr Gryll</speaker>
  <p>Bream, Doctor: there is not much to be said for bream.</p>
</sp>
<sp>
  <speaker>The Reverend Doctor Opimiam</speaker>
  <p>On the contrary, sir, I think there is much to be said for him.
     In the first place ...</p>
  <p>Fish, Miss Gryll — I could discourse to you on fish by the
     hour: but for the present I will forbear ...</p>
</sp>
```

Source: [153]

```
<sp>
  <speaker>Lord Curryfin</speaker>
  <stage>(after a pause).</stage>
  <p>
    <q>Mass</q> as the second grave-digger says
```

```
    in <title>Hamlet</title>, <q>I cannot tell.</q>
  </p>
</sp>
<p>A chorus of laughter dissolved the sitting.</p>
```

3.13 Overview of the Core Module

All the elements described in this chapter are provided by the core module.

Module core: Elements common to all TEI documents

- Elements defined: abbr add addrLine address analytic author bibl biblScope biblStruct binaryObject cb choice cit corr date del desc distinct divGen editor email emph expan foreign gap gloss graphic head headItem headLabel hi imprint index item l label lb lg list listBibl measure measureGrp meeting mentioned milestone monogr name note num orig p pb postBox postCode ptr pubPlace publisher q quote ref reg relatedItem resp respStmt rs said series sic soCalled sp speaker stage street teiCorpus term time title unclear

The selection and combination of modules to form a TEI schema is described in *1.2. Defining a TEI Schema*.

Chapter 4

Default Text Structure

This chapter describes the default high-level structure for TEI documents. A full TEI document combines metadata describing it, represented by a <teiHeader> element, with the document itself, represented by a <text> element. This basic pair is represented by a <TEI> element. The <teiHeader> element is specified by the header module, which is fully described in chapter *2. The TEI Header*. The remainder of the present chapter describes the <text> element and its high-level constituents.

A variant on this basic form, the <teiCorpus>, is also defined for the representation of language corpora, or other collections of encoded texts. A <teiCorpus> consists of one or more complete <TEI> elements, each combining a <teiHeader> and a <text> which itself carries a <teiHeader>. This permits the encoder to distinguish metadata applicable to the whole collection of encoded texts, which is represented by the outermost <teiHeader>, from that applicable to each of the individual <TEI> elements within the corpus. Further information about the organization and encoding of language corpora is given in chapter *15. Language Corpora*.

In summary, when the default structure module is included in a schema, the following elements are available for the representation of the outermost structure of a TEI document:

<TEI> (TEI document) contains a single TEI-conformant document, comprising a TEI header and a text, either in isolation or as part of a <teiCorpus> element.

<teiCorpus> contains the whole of a TEI encoded corpus, comprising a single corpus header and one or more TEI elements, each containing a single text header and a text.

<teiHeader> (TEI Header) supplies the descriptive and declarative information making up an electronic title page prefixed to every TEI-conformant text.

<text> contains a single text of any kind, whether unitary or composite, for example a poem or drama, a collection of essays, a novel, a dictionary, or a corpus sample.

As noted above, the <teiHeader> element is formally declared in the header module (see chapter *2. The TEI Header*). A TEI document may also contain elements from the model.resourceLike class (such as a collection of facsimile images, or a feature system declaration) if the appropriate module is included in a schema (see further *11.1. Digital Facsimiles* and *18.11. Feature System Declaration* respectively). By default, however, this class is not populated and hence only the elements <TEI>, <text>, and <teiCorpus> are available as major parts of a TEI document. These three elements are provided by the textstructure module described by the present chapter.

TEI texts may be regarded either as unitary, that is, forming an organic whole, or as composite, that is, consisting of several components which are in some important sense independent of each other. The distinction is not always entirely obvious: for example a collection of essays might be regarded as a single item in some circumstances, or as a number of distinct items in others. In such borderline cases, the encoder must choose whether to treat the text as unitary or composite; each may have advantages and disadvantages in a given situation.

Whether unitary or composite, the text is marked with the <text> tag and may contain front matter, a text body, and back matter. In unitary texts, the text body is tagged <body>; in composite texts, where the text body consists of a series of subordinate texts or groups, it is tagged <group>. The overall structure of any text, unitary or composite, is thus defined by the following elements:

<front> (front matter) contains any prefatory matter (headers, title page, prefaces, dedications, etc.) found at the start of a document, before the main body.

<body> (text body) contains the whole body of a single unitary text, excluding any front or back matter.

<group> contains the body of a composite text, grouping together a sequence of distinct texts (or groups of such texts) which are regarded as a unit for some purpose, for example the collected works of an author, a sequence of prose essays, etc.

<back> (back matter) contains any appendixes, etc. following the main part of a text.

The overall structure of a unitary text is:

```
<TEI>
  <teiHeader>
<!-- .... -->
  </teiHeader>
  <text>
    <front>
<!-- front matter of copy text, if any, goes here -->
    </front>
    <body>
<!-- body of copy text goes here -->
    </body>
    <back>
<!-- back matter of copy text, if any, goes here -->
    </back>
  </text>
</TEI>
```

The overall structure of a composite text made up of two unitary texts is:

```
<TEI>
  <teiHeader>
<!-- .... -->
  </teiHeader>
  <text>
    <front>
<!-- front matter for composite text -->
    </front>
    <group>
      <text>
        <front>
<!-- front matter of first unitary text, if any -->
        </front>
        <body>
<!-- body of first unitary text -->
        </body>
        <back>
<!-- back matter of first unitary text, if any -->
        </back>
      </text>
      <text>
```

```
      <body>
<!-- body of second unitary text -->
      </body>
     </text>
   </group>
   <back>
<!-- back matter for composite text, if any -->
   </back>
  </text>
</TEI>
```

Finally, a <floatingText> element is provided for the case where one text is embedded within another, but does not contribute to its hierarchical organization, for example because it interrupts it, or simply quoted within it. This is useful in such common literary contexts as the 'play within a play' or the narrative interrupted by other (often deeply nested) multiple narratives.

Each of these elements is further described in the remainder of this chapter. Elements <front> and <back> are further discussed in sections *4.5. Front Matter* and *4.7. Back Matter*. The <group> and <floatingText> elements, used for more complex or composite text structures, are further discussed in section *4.3. Grouped and Floating Texts*. Other textual elements, such as paragraphs, lists or phrases, which nest within these major structural elements, are discussed in chapter *3. Elements Available in All TEI Documents*, in the case of elements which can appear in any kind of document, or elsewhere in the case of elements specific to particular kinds of document.

4.1 Divisions of the Body

In some texts, the body consists simply of a sequence of low-level structural items, referred to here as components or component-level elements (see section *1.3. The TEI Class System*). Examples in prose texts include paragraphs or lists; in dramatic texts, speeches and stage directions; in dictionaries, dictionary entries. In other cases sequences of such elements will be grouped together hierarchically into textual divisions and subdivisions, such as chapters or sections. The names used for these structural subdivisions of texts vary with the genre and period of the text, or even at the whim of the author, editor, or publisher. For example, a major subdivision of an epic or of the Bible is generally called a 'book', that of a report is usually called a 'part' or 'section', that of a novel a 'chapter' — unless it is an epistolary novel, in which case it may be called a 'letter'. Even texts which are not organized as linear prose narratives, or not as narratives at all, will frequently be subdivided in a similar way: a drama into 'acts' and 'scenes'; a reference book into 'sections'; a diary or day book into 'entries'; a newspaper into 'issues' and 'sections', and so forth.

Because of this variety, these Guidelines propose that all such textual divisions be regarded as occurrences of the same neutrally named elements, with an attribute **type** used to categorize elements independently of their hierarchic level. Two alternative styles are provided for the marking of these neutral divisions: numbered and un-numbered. Numbered divisions are named <div1>, <div2>, etc., where the number indicates the depth of this particular division within the hierarchy, the largest such division being 'div1', any subdivision within it being 'div2', any further sub-sub-division being 'div3' and so on. Un-numbered divisions are simply named , and allowed to nest recursively to indicate their hierarchic depth. The two styles must *not* be combined within a single <front>, <body>, or <back> element.

4.1.1 Un-numbered Divisions

The following element is used to identify textual subdivisions in the un-numbered style:

** ** (text division) contains a subdivision of the front, body, or back of a text.

As a member of the class att.typed, this element has the following additional attributes:

att.typed provides attributes which can be used to classify or subclassify elements in any way.

@**type** characterizes the element in some sense, using any convenient classification scheme or typology.

@**subtype** provides a sub-categorization of the element, if needed

Using this style, the body of a text containing two parts, each composed of two chapters, might be represented as follows:

```
<body>
  <div type="part" n="1">
    <div type="chapter" n="1">
<!-- text of part 1, chapter 1 -->
    </div>
    <div type="chapter" n="2">
<!-- text of part 1, chapter 2 -->
    </div>
  </div>
  <div type="part" n="2">
    <div n="1" type="chapter">
<!-- text of part 2, chapter 1 -->
    </div>
    <div n="2" type="chapter">
<!-- text of part 2, chapter 2 -->
    </div>
  </div>
</body>
```

4.1.2 Numbered Divisions

The following elements are used to identify textual subdivisions in the numbered style:

<div1> (level-1 text division) contains a first-level subdivision of the front, body, or back of a text.

<div2> (level-2 text division) contains a second-level subdivision of the front, body, or back of a text.

<div3> (level-3 text division) contains a third-level subdivision of the front, body, or back of a text.

<div4> (level-4 text division) contains a fourth-level subdivision of the front, body, or back of a text.

<div5> (level-5 text division) contains a fifth-level subdivision of the front, body, or back of a text.

<div6> (level-6 text division) contains a sixth-level subdivision of the front, body, or back of a text.

<div7> (level-7 text division) contains the smallest possible subdivision of the front, body or back of a text, larger than a paragraph.

As members of the class att.typed these elements all bear the following additional attributes:

att.typed provides attributes which can be used to classify or subclassify elements in any way.

@**type** characterizes the element in some sense, using any convenient classification scheme or typology.

@**subtype** provides a sub-categorization of the element, if needed

The largest possible subdivision of the body is <div1> element and the smallest possible <div7>. If numbered divisions are in use, a division at any one level (say, <div3>), may contain only numbered divisions at the next lowest level (in this case, <div4>).

Using this style, the body of a text containing two parts, each composed of two chapters, might be represented as follows:

```
<body>
  <div1 type="part" n="1">
    <div2 type="chapter" n="1">
<!-- text of part 1, chapter 1 -->
    </div2>
    <div2 type="chapter" n="2">
<!-- text of part 1, chapter 2 -->
    </div2>
  </div1>
  <div1 type="part" n="2">
    <div2 n="1" type="chapter">
<!-- text of part 2, chapter 1 -->
    </div2>
    <div2 n="2" type="chapter">
<!-- text of part 2, chapter 2 -->
    </div2>
  </div1>
</body>
```

4.1.3 Numbered or Un-numbered?

Within the same <front>, <body>, or <back> element, all hierarchic subdivisions must be marked using either nested elements, or <div1>, <div2> etc. elements nested as appropriate; the two styles must *not* be mixed.

The choice between numbered and un-numbered divisions will depend to some extent on the complexity of the material: un-numbered divisions allow for an arbitrary depth of nesting, while numbered divisions limit the depth of the tree which can be constructed. Where divisions at different levels should be processed differently (for example to ensure that chapters, but not sections, begin on a new page), numbered divisions slightly simplify the task of defining the desired processing for each level, though this distinction could also be made by supplying this information on the **type** attribute of an un-numbered . Some software may find numbered divisions easier to process, as there is no need to maintain knowledge of the whole document structure in order to know the level at which a division occurs; such software may, however, find it difficult to cope with some other aspects of the TEI scheme. On the other hand, in a collection of many works it may prove difficult or impossible to ensure that the same numbered division always corresponds with the same type of textual feature: a 'chapter' may be at level 1 in one work and level 3 in another.

Whichever style is used, the global **n** and **xml:id** attributes (section *1.3.1.1. Global Attributes*) may be used to provide reference strings or labels for each division of a text, where appropriate. Such labels should be provided for each section which is regarded as significant for referencing purposes (on reference systems, see further section *3.10. Reference Systems*).

As indicated above, the **type** and **subtype** attributes provided by the att.typed class may be used to provide a name or description for the division. Typical values might be 'book', 'chapter', 'section', 'part', or (for verse texts) 'book', 'canto', 'stanza', or (for dramatic texts) 'act', 'scene'. The following extended example uses numbered divisions to indicate the structure of a novel, and illustrates the use of the attributes discussed above. It also uses some elements discussed in section *4.2. Elements Common to All Divisions* and the <p> element discussed in section *3.1. Paragraphs*.

```
<div1 type="book" n="I" xml:id="JA0100">
  <head>Book I.</head>
  <div2 type="chapter" n="1" xml:id="JA0101">
    <head>Of writing lives in general, and particularly of Pamela, with a word
        by the bye of Colley Cibber and others.</head>
```

```
   <p>It is a trite but true observation, that examples work more forcibly on
       the mind than precepts: ... </p>
<!-- remainder of chapter 1 here -->
  </div2>
  <div2 type="chapter" n="2" xml:id="JA0102">
   <head>Of Mr. Joseph Andrews, his birth, parentage, education, and great
       endowments; with a word or two concerning ancestors.</head>
   <p>Mr. Joseph Andrews, the hero of our ensuing history, was esteemed to
       be the only son of Gaffar and Gammar Andrews, and brother to the
       illustrious Pamela, whose virtue is at present so famous ... </p>
<!-- remainder of chapter 2 here -->
  </div2>
<!-- remaining chapters of Book 1 here -->
  <trailer>The end of the first Book</trailer>
</div1>
<div1 type="book" n="II" xml:id="JA0200">
  <head>Book II</head>
  <div2 type="chapter" n="1" xml:id="JA0201">
   <head>Of divisions in authors</head>
   <p>There are certain mysteries or secrets in all trades, from the highest
       to the lowest, from that of <term>prime-ministering</term>, to this of
   <term>authoring</term>, which are seldom discovered unless to members of
       the same calling ... </p>
   <p>I will dismiss this chapter with the following observation: that it
       becomes an author generally to divide a book, as it does a butcher to
       joint his meat, for such assistance is of great help to both the reader
       and the carver. And now having indulged myself a little I will endeavour
       to indulge the curiosity of my reader, who is no doubt impatient to know
       what he will find in the subsequent chapters of this book.</p>
  </div2>
  <div2 type="chapter" n="2" xml:id="JA0202">
   <head>A surprising instance of Mr. Adams's short memory, with the
       unfortunate consequences which it brought on Joseph.
   </head>
   <p>Mr. Adams and Joseph were now ready to depart different ways ... </p>
  </div2>
</div1>
```

Source: [71]

As an alternative (or complement) to this use of the **type** attribute to characterize neutrally named division elements, the modification mechanisms discussed in section *23.2. Personalization and Customization* may be used to define new elements such as <chapter>, <part>, etc. To make this simpler, a single member model class is defined for each of the neutrally named division elements: model.divLike (containing), model.div1Like (containing <div1>), model.div2Like (containing <div2>), etc. For example, suppose that the body of a text consists of a series of diary entries, each of which is potentially divided into entries for the morning and the afternoon. This might be represented in any of the following ways. First, using the un-numbered style:

```
<body>
  <div type="entry" n="1">
   <div type="morning" n="1.1">
    <p>....</p>
   </div>
   <div type="afternoon" n="1.2">
    <p>....</p>
```

```
    </div>
  </div>
  <div type="entry" n="2">
    <div type="morning" n="2.1">
      <p>....</p>
    </div>
    <div type="afternoon" n="2.2">
      <p>....</p>
    </div>
  </div>
<!-- ...-->
</body>
```

Equivalently, using the numbered style:

```
<body>
  <div1 type="entry" n="1">
    <div2 type="morning" n="1.1">
      <p>....</p>
    </div2>
    <div2 type="afternoon" n="1.2">
      <p>....</p>
    </div2>
  </div1>
  <div1 type="entry" n="2">
    <div2 type="morning" n="2.1">
      <p>....</p>
    </div2>
    <div2 type="afternoon" n="2.2">
      <p>....</p>
    </div2>
  </div1>
<!-- ...-->
</body>
```

Now, assuming a customization in which a new element <diaryEntry> has been added to the model.divLike class:

```
<body
   xmlns:my="http://www.example.org/ns/nonTEI">
  <diaryEntry xmlns="http://www.example.org/ns/nonTEI"
    type="entry" n="1">
    <diaryEntry xmlns="http://www.example.org/ns/nonTEI"
      type="morning" n="1.1">
      <p>....</p>
    </diaryEntry>
    <diaryEntry xmlns="http://www.example.org/ns/nonTEI"
      type="afternoon" n="1.2">
      <p>....</p>
    </diaryEntry>
  </diaryEntry>
  <diaryEntry xmlns="http://www.example.org/ns/nonTEI"
    type="entry" n="1">
    <diaryEntry xmlns="http://www.example.org/ns/nonTEI"
```

```
     type="morning"  n="1.1">
     <p>....</p>
   </diaryEntry>
   <diaryEntry xmlns="http://www.example.org/ns/nonTEI"
     type="afternoon"  n="1.2">
     <p>....</p>
   </diaryEntry>
  </diaryEntry>
<!-- ...-->
</body>
```

And finally, assuming a customization in which three new elements have been added: <diaryEntry> to the model.div1 class, and <amEntry> and <pmEntry> both to the model.div2 class:

```
<body
   xmlns:my="http://www.example.org/ns/nonTEI">
  <diaryEntry xmlns="http://www.example.org/ns/nonTEI"
    type="entry"  n="1">
    <amEntry xmlns="http://www.example.org/ns/nonTEI"
      type="morning"  n="1.1">
      <p>....</p>
    </amEntry>
    <pmEntry xmlns="http://www.example.org/ns/nonTEI"
      type="afternoon"  n="1.2">
      <p>....</p>
    </pmEntry>
  </diaryEntry>
  <diaryEntry xmlns="http://www.example.org/ns/nonTEI"
    type="entry"  n="1">
    <amEntry xmlns="http://www.example.org/ns/nonTEI"
      type="morning"  n="1.1">
      <p>....</p>
    </amEntry>
    <pmEntry xmlns="http://www.example.org/ns/nonTEI"
      type="afternoon"  n="1.1">
      <p>....</p>
    </pmEntry>
  </diaryEntry>
<!-- ... -->
</body>
```

More information about the customization techniques exemplified here is provided in *23.2. Personalization and Customization*.

4.1.4 Partial and Composite Divisions

In most situations, the textual subdivisions marked by or <div1> (etc.) elements will be both complete and identically organized with reference to the original source. For some purposes however, in particular where dealing with unusually large or unusually small texts, encoders may find it convenient to present as textual divisions sequences of text which are incomplete with reference to the original text, or which are in fact an ad hoc agglomeration of tiny texts. Moreover, in some kinds of texts it is difficult or impossible to determine the order in which individual subdivisions should be combined to form the next higher level of subdivision, as noted below.

To overcome these problems, the following additional attributes are defined for all elements in the att.divLike class:

att.divLike provides attributes common to all elements which behave in the same way as divisions.

@**org** (organization) specifies how the content of the division is organized.

@**sample** indicates whether this division is a sample of the original source and if so, from which part.

@**part** specifies whether or not the division is fragmented by some other structural element, for example a speech which is divided between two or more verse stanzas.

For example, an encoder might choose to transcribe only the first two thousand words of each chapter from a novel. In such a case, each chapter might conveniently be regarded as a partial division, and tagged with a element in the following form:

```
<div
   n="xx"
   sample="initial"
   part="Y"
   type="chapter">
  <p> ... </p>
</div>
```

where **xx** represents a number for the chapter, and the **part** attribute takes the value **Y** to indicate that this division is incomplete in some respect. Other possible values for this attribute indicate whether material has been omitted at the end (F), the beginning (B), or in the middle (M) of the division, while the <gap> element (*3.4.3. Additions, Deletions, and Omissions*) may be used to indicate exactly where material has been omitted:

```
<div  n="xx"  part="M"  type="chapter">
  <p> ... </p>
  <gap  extent="2"  reason="sampling"/>
  <p> ... </p>
</div>
```

The <samplingDecl> element in the TEI Header should also be used to record the principles underlying the selection of incomplete samples, as further described in section *2.3.2. The Sampling Declaration*.

The following example demonstrates how a newspaper column composed of very short unrelated snippets may be encoded using these attributes:

```
<div1  type="storylist"  org="composite">
  <head>News in brief</head>
  <div2  type="story">
    <head>Police deny <soCalled>losing</soCalled> bomb</head>
    <p>Scotland Yard yesterday denied claims in the Sunday
        Express that anti-terrorist officers trailing an IRA van
        loaded with explosives in north London had lost track of
        it 10 days ago.</p>
  </div2>
  <div2  type="story">
    <head>Hotel blaze</head>
    <p>Nearly 200 guests were evacuated before dawn
        yesterday after fire broke out at the Scandic
        Crown hotel in the Royal Mile, Edinburgh.</p>
```

```
  </div2>
  <div2  type="story">
    <head>Test match split</head>
    <p>Test Match Special next summer will be split
        between Radio 5 and Radio 3, after protests this
        year that it disrupted Radio 3's music schedule.</p>
  </div2>
</div1>
```

Source: [193]

The **org** attribute on the <div1> element is used here to indicate that individual stories in this group, marked here as <div2>, are really quite independent of each other, although they are all marked as subdivisions of the whole group. They can be read in any order without affecting the sense of the piece; indeed, in some cases, divisions of this nature are printed in such a way as to make it impossible to determine the order in which they are intended to be read. Individual stories can be added or removed without affecting the existing components.

This method of encoding composite texts as composite divisions has some limitations compared with the more general and powerful mechanisms discussed in section *4.3.1. Grouped Texts*. However, it may be preferable in some circumstances, notably where the individual texts are very small.

4.2 Elements Common to All Divisions

The divisions of any kind of text may sometimes begin with a brief heading or descriptive title, with or without a byline, an epigraph or brief quotation, or a salutation such as one finds at the start of a letter. They may also conclude with a brief trailer, byline, postscript, or signature. Many of these (e.g. a byline) may appear either at the start or at the end of a text division proper.

To support this heterogeneity, the TEI architecture defines five classes, all of which are populated by this module:

model.divTop groups elements appearing at the beginning of a text division.

model.divTopPart groups elements which can occur only at the beginning of a text division.

model.divBottom groups elements appearing at the end of a text division.

model.divBottomPart groups elements which can occur only at the end of a text division.

model.divWrapper groups elements which can appear at either top or bottom of a textual division.

By default the class model.divWrapper provides the following special-purpose elements:

<argument> A formal list or prose description of the topics addressed by a subdivision of a text.

<byline> contains the primary statement of responsibility given for a work on its title page or at the head or end of the work.

<dateline> contains a brief description of the place, date, time, etc. of production of a letter, newspaper story, or other work, prefixed or suffixed to it as a kind of heading or trailer.

<docAuthor> (document author) contains the name of the author of the document, as given on the title page (often but not always contained in a byline).

<docDate> (document date) contains the date of a document, as given (usually) on a title page.

<epigraph> contains a quotation, anonymous or attributed, appearing at the start of a section or chapter, or on a title page.

The class model.divTop combines these elements with the following elements, which populate the model.divTopPart class:

<head> (heading) contains any type of heading, for example the title of a section, or the heading of a list, glossary, manuscript description, etc.

<salute> (salutation) contains a salutation or greeting prefixed to a foreword, dedicatory epistle, or other division of a text, or the salutation in the closing of a letter, preface, etc.

<opener> groups together dateline, byline, salutation, and similar phrases appearing as a preliminary group at the start of a division, especially of a letter.

For further details of the <head> element, see section *4.2.1. Headings and Trailers*; for <epigraph> and <argument>, see section *4.2.3. Arguments, Epigraphs, and Postscripts*; for <opener>, see section *4.2.2. Openers and Closers*.

The class model.divBottom combines these elements with the following elements, which populate the model.divBottomPart class:

<closer> groups together salutations, datelines, and similar phrases appearing as a final group at the end of a division, especially of a letter.

<signed> (signature) contains the closing salutation, etc., appended to a foreword, dedicatory epistle, or other division of a text.

<trailer> contains a closing title or footer appearing at the end of a division of a text.

<postscript> contains a postscript, e.g. to a letter.

For further details of the <trailer> element, see section *4.2.1. Headings and Trailers*; for the <closer> and <signed> elements, section *4.2.2. Openers and Closers*; for the <postscript> element, section *4.2.3. Arguments, Epigraphs, and Postscripts*.

4.2.1 Headings and Trailers

The <head> element is used to identify a heading prefixed to the start of any textual division, at any level. A given division may contain more than one such element, as in the following example:

```
<div1  n="Etym">
  <head>Etymology</head>
  <head>(Supplied by a late consumptive usher to a
     grammar school)</head>
  <p>The pale Usher — threadbare in coat, heart,
     body and brain; I see him now. He was ever
     dusting his old lexicons and grammars, ...</p>
</div1>
```

Source: [140]

Unlike some other markup schemes, the TEI scheme does *not* require that headings attached to textual subdivisions at different hierarchic levels have different identifiers. All kinds of heading are marked identically using the <head> tag; the type or level of heading intended is implied by the immediate parent of the <head> element, which may for example be a <div1>, <div2>, etc., an un-numbered , or any member of the model.listLike class. However, as with elements, the encoder may choose to extend the model.headLike class of which <head> is the sole member to include other such elements if required.

In certain kinds of text (notably newspapers), there may be a need to categorize individual headings within the sequence at the start of a division, for example as 'main' headings, or 'detail' headings: this may readily be done using the **type** or **subtype** attribute. Specific elements are provided for certain kinds of heading-like features, (notably <byline>, <dateline>, and <salute>; see further section *4.2.2. Openers and Closers*), but the **type** or **subtype** attributes must be used to discriminate among other forms of heading. These attributes are provided, as elsewhere, by the att.typed attribute class of which the <head> element is a member.

In the following example, taken from a British newspaper, the lead story and its associated headlines have been encoded as a element, with appropriate model.divTop elements attached:

```
<div type="story">
  <head rend="underlined" type="sub">President pledges safeguards for 2,400 British
     troops in Bosnia</head>
  <head rend="scream" type="main">Major agrees to enforced no-fly zone</head>
  <byline>By George Jones, Political Editor, in Washington</byline>
  <p>Greater Western intervention in the conflict in
     former Yugoslavia was pledged by President Bush ...</p>
</div>
```

Source: [50]

In older writings, the headings or incipits may be longer than in modern works. When heading-like material appears in the middle of a text, the encoder must decide whether or not to treat it as the start of a new division. If the phrase in question appears to be more closely connected with what follows than with what precedes it, then it may be regarded as a heading and tagged as the <head> of a new element. If it appears to be simply inserted or superimposed — as for example the kind of 'pull quotes' often found in newspapers or magazines, then the <quote>, <q>, or <cit> element may be more appropriate.

The <trailer> element, which can appear at the end of a division only, is used to mark any heading-like feature appearing in this position, as in this example:

```
<div type="book" n="I">
  <head>In the name of Christ here begins the
     first book of the ecclesiastical history of Georgius Florentinus,
     known as Gregory, Bishop of Tours.</head>
  <div>
    <head>Chapter Headings</head>
    <list>
<!-- list of chapter heads omitted -->
    </list>
  </div>
  <div>
    <head>In the name of Christ here begins Book I of the history.</head>
    <p>Proposing as I do ...</p>
    <p>From the Passion of our Lord until the death of Saint Martin four
        hundred and twelve years passed.</p>
    <trailer>Here ends the first Book, which covers five thousand, five
        hundred and ninety-six years from the beginning of the world down
        to the death of Saint Martin.</trailer>
  </div>
</div>
```

Source: [92]

4.2.2 Openers and Closers

In addition to headings of various kinds, divisions sometimes include more or less formulaic opening or closing passages, typically conveying such information as the name and address of the person to whom the division is addressed, the place or time of its production, a salutation or exhortation to the reader, and so on. Divisions in epistolary form are particularly liable to include such features. Additional elements for the detailed encoding of personal names, dates, and places are provided in chapter *13. Names, Dates, People, and Places*. For simple cases, the following elements should be adequate:

<byline> contains the primary statement of responsibility given for a work on its title page or at the head or end of the work.

<dateline> contains a brief description of the place, date, time, etc. of production of a letter, newspaper story, or other work, prefixed or suffixed to it as a kind of heading or trailer.

<salute> (salutation) contains a salutation or greeting prefixed to a foreword, dedicatory epistle, or other division of a text, or the salutation in the closing of a letter, preface, etc.

<signed> (signature) contains the closing salutation, etc., appended to a foreword, dedicatory epistle, or other division of a text.

The <byline> and <dateline> elements are used to encode headings which identify the authorship and provenance of a division. Although the terminology derives from newspaper usage, there is no implication that <dateline> or <byline> elements apply only to newspaper texts. The following example illustrates use of the <dateline> and <signed> elements at the end of the preface to a novel:

```
<div type="preface">
  <head>To Henry Hope.</head>
  <p>It is not because this volume was conceived and partly
     executed amid the glades and galleries of the Deepdene,
     that I have inscribed it with your name. ... I shall find a
     reflex to their efforts in your own generous spirit and
     enlightened mind.
  </p>
  <closer>
    <signed xml:lang="el">D.</signed>
    <dateline>Grosvenor Gate, May-Day, 1844</dateline>
  </closer>
</div>
```

Source: [61]

Where a sequence of such elements appear together, either at the beginning or end of an element, it may be convenient to group them together using one of the following elements:

<opener> groups together dateline, byline, salutation, and similar phrases appearing as a preliminary group at the start of a division, especially of a letter.

<closer> groups together salutations, datelines, and similar phrases appearing as a final group at the end of a division, especially of a letter.

The following examples demonstrate the use of the <opener> and <closer> grouping elements:

```
<div type="narrative" n="6">
  <head>Sixth Narrative</head>
  <head>contributed by Sergeant Cuff</head>
  <div type="fragment" n="6.1">
    <opener>
      <dateline>
        <name type="place">Dorking, Surrey,</name>
        <date>July 30th, 1849</date>
      </dateline>
      <salute>To <name>Franklin Blake, Esq.</name> Sir, —</salute>
    </opener>
    <p>I beg to apologize for the delay that has occurred in the
        production of the Report, with which I engaged to furnish you.
        I have waited to make it a complete Report ...</p>
    <closer>
      <salute>I have the honour to remain, dear sir, your
           obedient servant </salute>
```

```
     <signed>
       <name>RICHARD CUFF</name> (late sergeant in the
           Detective Force, Scotland Yard, London). </signed>
    </closer>
  </div>
</div>
```

Source: [47]

```
<div  type="letter"  n="14">
  <head>Letter XIV: Miss Clarissa Harlowe to Miss Howe</head>
  <opener>
    <dateline>Thursday evening, March 2.</dateline>
  </opener>
  <p>On Hannah's depositing my long letter ...</p>
  <p>An interruption obliges me to conclude myself
     in some hurry, as well as fright, what I must ever be,</p>
  <closer>
    <salute>Yours more than my own,</salute>
    <signed>Clarissa Harlowe</signed>
  </closer>
</div>
```

Source: [164]

For further discussion of the encoding of dates and of names of persons and places, see section *3.5.4. Dates and Times* and chapter *13. Names, Dates, People, and Places.*

4.2.3 Arguments, Epigraphs, and Postscripts

The <argument> element may be used to encode the prefatory list of topics sometimes found at the start of a chapter or other division. It is most conveniently encoded as a list, since this allows each item to be distinguished, but may also simply be presented as a paragraph. The following are thus both equally valid ways of encoding the same argument:

```
<div  type="chap"  n="6">
  <argument>
    <p>Kingston — Instructive remarks on early English history
        — Instructive observations on carved oak and life in general
        — Sad case of Stivvings, junior — Musings on antiquity
        — I forget that I am steering — Interesting result
        — Hampton Court Maze — Harris as a guide.</p>
  </argument>
  <p>It was a glorious morning, late spring or early summer, as you
     care to take it ...</p>
</div>
```

Source: [106]

```
<div  type="chap"  n="6">
  <argument>
    <list  type="inline">
      <item>Kingston</item>
```

```
     <item>Instructive remarks on early English history</item>
     <item>Instructive observations on carved oak and life in
           general</item>
     <item>Sad case of Stivvings, junior</item>
     <item>Musings on antiquity</item>
     <item>I forget that I am steering</item>
     <item>Interesting result</item>
     <item>Hampton Court Maze</item>
     <item>Harris as a guide.</item>
   </list>
  </argument>
  <p>It was a glorious morning, late spring or early summer, as you
    care to take it ...</p>
</div>
```

Source: [106]

An epigraph is a quotation from some other work appearing on a title page, or at the start of a division. It may be encoded using the special-purpose <epigraph> element. Its content will generally be a <q> or <quote> element, often associated with a bibliographic reference, as in the following example:

```
<div  n="19"  type="chap">
  <head>Chapter 19</head>
  <epigraph>
    <cit>
     <quote>I pity the man who can travel
           from Dan to Beersheba, and say <q>'Tis all
              barren;</q> and so is all the world to him
           who will not cultivate the fruits it offers.
     </quote>
     <bibl>Sterne: Sentimental Journey.</bibl>
    </cit>
  </epigraph>
  <p>To say that Deronda was romantic would be to
    misrepresent him: but under his calm and somewhat
    self-repressed exterior ...</p>
</div>
```

Source: [69]

For discussion of quotations appearing other than as epigraphs refer to section *3.3.3. Quotation*.

A postscript is a passage added after the signature of a letter or, less frequently, the main portion of the body of a book, article, or essay. In English a postscript is often abbreviated as *P.S.* or *PS*, and postscripts are often introduced by labels with one of these abbreviations, as in the following example.

```
<div  type="letter">
  <opener>
    <dateline>
     <placeName>Newport</placeName>
     <date  when="1761-05-27">May ye 27th 1761</date>
    </dateline>
    <salute>Gentlemen</salute>
  </opener>
  <p>Capt Stoddard's Business
  <lb/>calling him to Providence, have
```

```
  <lb/>got him to look at Hopkins brigantine
  <lb/>& if can agree to Purchase her, shall
  <lb/>be much oblig'd for your further
  <lb/>assistance herein, & will acquiesce with
  <lb/>whatever you & he shall Contract
  <lb/>for — I Thank you for your
  <lb/>
    <unclear>Line</unclear> respecting the brigantine & Beg
  <lb/>leave to Recommend the Bearer
  <lb/>to you for your advice & Friendship
  <lb/>in this matter</p>
  <closer>
    <salute>I am your most humble servant</salute>
    <signed>Joseph Wanton Jr</signed>
  </closer>
  <postscript>
    <label>P.S.</label>
    <p>I have Mollases, Sugar,
    <lb/>Coffee & Rum, which
    <lb/>will Exchange with you
    <lb/>for Candles or Oyl</p>
  </postscript>
</div>
```

Source: [204]

4.2.4 Content of Textual Divisions

Other than elements from the model.divWrapper, model.divTop, or model.divBottom classes, every textual division (numbered or un-numbered) consists of a sequence of ungrouped macro.component elements (see *1.3. The TEI Class System*). The actual elements available will depend on the modules in use; in all cases, at least the component-level structural elements defined in the core will be available (paragraphs, lists, dramatic speeches, verse lines and line groups etc.). If the drama module has been selected, then other component- or phrase- level items specialised for performance texts (for example, cast lists or camera angles) will be available, as defined in chapter *7. Performance Texts*) will be available. If the dictionary module is in use, then dictionary entries, related entries, etc. (as defined in chapter *9. Dictionaries*) will also be available; if the module for transcribed speech is in use, then utterances, pauses, vocals, kinesics, etc., as defined in chapter *8.3. Elements Unique to Spoken Texts* will be available; and so on.

Where a text contains low-level elements from more than one module these may appear at any point; there is no requirement that elements from the same module be kept together.

4.3 Grouped and Floating Texts

The <group> element discussed in *4.3.1. Grouped Texts* should be used to represent a collection of independent texts which is to be regarded as a single unit for processing or other purposes. The <floatingText> element discussed in *4.3.2. Floating Texts* should be used to represent an independent text which interrupts the text containing it at any point but after which the surrounding text resumes.

<group> contains the body of a composite text, grouping together a sequence of distinct texts (or groups of such texts) which are regarded as a unit for some purpose, for example the collected works of an author, a sequence of prose essays, etc.

<floatingText> contains a single text of any kind, whether unitary or composite, which interrupts the text containing it at any point and after which the surrounding text resumes.

4.3.1 Grouped Texts

Examples of composite texts which should be represented using the <group> element include anthologies and other collections. The presence of common front matter referring to the whole collection, possibly in addition to front matter relating to each individual text, is a good indication that a given text might usefully be encoded in this way; this structure may be found useful in other circumstances too.

For example, the overall structure of a collection of short stories might be encoded as follows:

```
<TEI>
  <teiHeader>
<!-- header information for the whole collection -->
  </teiHeader>
  <text>
    <front>
      <docTitle>
        <titlePart> The Adventures of Sherlock Holmes
        </titlePart>
      </docTitle>
      <docImprint>First published in <title>The Strand</title>
            between July 1891 and December 1892</docImprint>
<!-- any other front matter specific to this collection -->
    </front>
    <group>
      <text>
        <front>
          <head  rend="italic">Adventures of Sherlock
                   Holmes</head>
          <docTitle>
            <titlePart>Adventure I. —</titlePart>
            <titlePart>A Scandal in Bohemia</titlePart>
          </docTitle>
          <byline>By A. Conan Doyle.</byline>
        </front>
        <body>
          <p>To Sherlock Holmes she is always
          <emph>the</emph> woman. ... </p>
<!-- remainder of A Scandal in Bohemia here -->
        </body>
      </text>
      <text>
        <front>
          <head  rend="italic">Adventures of Sherlock Holmes</head>
          <docTitle>
            <titlePart>Adventure II. —</titlePart>
            <titlePart>The Red-Headed League</titlePart>
          </docTitle>
          <byline>By A. Conan Doyle.</byline>
        </front>
        <body>
<!-- text of The Red Headed League here -->
        </body>
      </text>
      <text>
        <front>
          <head  rend="italic">Adventures of Sherlock Holmes</head>
          <docTitle>
            <titlePart>Adventure XII. —</titlePart>
```

```
          <titlePart>The Adventure of the Copper Beeches</titlePart>
        </docTitle>
        <byline>By A. Conan Doyle.</byline>
      </front>
      <body>
        <p>
          <q>To the man who loves art for its
                    own sake,</q> remarked Sherlock Holmes ...


<!-- remainder of The Copper Beeches here -->

                  ... she is now the head of a private school
                  at Walsall, where I believe that she has
                  met with considerable success.</p>
      </body>
    </text>
<!-- end of The Copper Beeches -->
   </group>
  </text>
<!-- end of the Adventures of Sherlock Holmes -->
</TEI>
```

Source: [63]

A text which is a member of a group may itself contain groups. This is quite common in collections of verse, but may happen in any kind of text. As an example, consider the overall structure of a typical collection, such as the *Muses Library* edition of Crashaw's poetry. Following a critical introduction and table of contents, this work contains the following major sections:

- *Steps to the Temple* (a collection of verse first published in 1648)
- *Carmen deo Nostro* (a second collection, published in 1652)
- *The Delights of the Muses* (a third collection, published in 1648)
- *Posthumous Poems,* I (a collection of fragments all taken from a single manuscript)
- *Posthumous Poems,* II (a further collection of fragments, taken from a different manuscript)

Each of the three collections published in Crashaw's lifetime has a reasonable claim to be considered as a text in its own right, and may therefore be encoded as such. It is rather more arbitrary as to whether the two posthumous collections should be treated as two groups, following the practice of the *Muses Library* edition. An encoder might elect to combine the two into a single group or simply to treat each fragment as an ungrouped unitary text.

The *Muses Library* edition reprints the whole of each of the three original collections, including their original front matter (title pages, dedications etc.). These should be encoded using the <front> element and its constituents (on which see further section *4.5. Front Matter*), while the body of each collection should be encoded as a single <group> element. Each individual poem within the collections should be encoded as a distinct <text> element. The beginning of the whole collection would thus appear as follows (for further discussion of the use of the elements and <lg> for textual subdivision of verse, see section *3.12.1. Core Tags for Verse* and chapter *6. Verse*):

```
<text>
  <front>
    <titlePage>
```

```
    <docTitle>
      <titlePart>The poems of Richard Crashaw</titlePart>
    </docTitle>
    <byline>Edited by J.R. Tutin</byline>
  </titlePage>
  <div  type="preface">
    <head>Editor's Note</head>
    <p>A few words are necessary ... </p>
  </div>
</front>
<group>
  <text>
    <front>
      <titlePage>
        <docTitle>
          <titlePart>Steps to the Temple, Sacred Poems</titlePart>
        </docTitle>
      </titlePage>
      <div  type="address">
        <head>The Preface to the Reader</head>
        <p>Learned Reader, The Author's friend will not usurp much
                  upon thy eye ... </p>
      </div>
    </front>
    <group>
      <text>
        <front>
          <docTitle>
            <titlePart>Sospetto D'Herode</titlePart>
          </docTitle>
        </front>
        <body>
          <div1  type="book"  n="Herod I">
            <head>Libro Primo</head>
            <epigraph>
              <l>Casting the times with their strong signs</l>
            </epigraph>
            <lg  n="I.1"  type="stanza">
              <l>Muse!  now the servant of soft loves no more</l>
              <l>Hate is thy theme and Herod whose unblest</l>
              <l>Hand (O, what dares not jealous greatness?)  tore</l>
              <l>A thousand sweet babes from their mothers' breast,</l>
              <l>The blooms of martyrdom ...</l>
            </lg>
          </div1>
        </body>
      </text>
      <text>
        <front>
          <docTitle>
            <titlePart>The Tear</titlePart>
          </docTitle>
        </front>
        <body>
          <lg  n="I">
            <l>What bright soft thing is this</l>
            <l>Sweet Mary, thy fair eyes' expense?</l>
```

```
          </lg>
        </body>
      </text>
<!-- remaining poems of the Steps to the Temple appear here, each tagged as a distinct text element -->
     </group>
     <back>
<!-- back matter for the Steps to the Temple -->
     </back>
   </text>
   <text>
<!-- start of Carmen deo Nostro -->
     <front/>
     <group>
       <text/>
       <text/>
<!-- more texts here -->
     </group>
   </text>
   <text>
<!-- start of The Delights of the Muses -->
     <group>
       <text/>
       <text/>
<!-- more texts here -->
     </group>
   </text>
  </group>
  <back>
<!-- back matter for the whole collection -->
  </back>
</text>
```

Source: [49]

The <group> element may be used in this way to encode any kind of collection of which the constituents are regarded by the encoder as texts in their own right. Examples include anthologies or collections of verse or prose by multiple authors, florilegia, or commonplace books, journals, day books, etc. As a fairly typical example, we consider *The Norton Book of Travel*, an anthology edited by Paul Fussell and published in 1987 by W. W. Norton. This work comprises the following major sections:

1. Front matter (title page, acknowledgments, introductory essay)
2. The Beginnings
3. The Eighteenth Century and the Grand Tour
4. The Heyday
5. Touristic Tendencies
6. Post Tourism
7. Back matter (permissions list, index)

Each titled section listed above comprises a group of extracts or complete texts from writers of a given historical period, preceded by an introductory essay. For example, the second group listed above contains, inter alia, the following:

1. Prefatory essay
2. Five letters by Lady Mary Wortley Montagu
3. An extract from Swift's Gullivers Travels
4. Two poems by Alexander Pope
5. Two extracts from Boswell's Journal
6. A poem by William Blake

Each group of writings by a single author is preceded by a brief biographical notice. Some of the extracts are quite lengthy, containing several chapters or other divisions; others are quite short. As the above list indicates, the texts included range across all kinds of material: verse, prose, journals and letters.

The easiest way of encoding such an anthology is to treat each individual extract as a text in its own right. A sequence of texts by a single author, together with the biographical note preceding it, can then be treated as a single <group> element within the larger <group> formed by the section. The sequence of single or composite texts making up a single section of the work is likewise treated, together with its prefatory essay, as a single <group> within the work. Schematically:

```
<text>
<!-- the whole anthology -->
  <front>
<!-- title page, acknowledgments, introductory essay -->
  </front>
  <group>
<!-- body of anthology starts here -->
    <group>
      <head>The Beginnings</head>
<!-- sequence of texts or groups -->
    </group>
    <group>
<!-- The Eighteenth Century and the Grand Tour -->
      <text>
<!-- prefatory essay by editor -->
      </text>
      <group>
<!-- Section on Lady Mary Wortley Montagu starts -->
        <text>
<!-- biographical notice by editor -->
        </text>
        <text>
<!-- first letter -->
        </text>
        <text>
<!-- second letter -->
        </text>
<!-- ... -->
      </group>
<!-- end of Montagu section -->
      <text>
<!-- single text by Jonathan Swift starts -->
        <front>
<!-- biographical notice by editor -->
        </front>
```

```
      <body/>
     </text>
<!-- end of Swift section -->
     <group>
<!-- Section on Alexander Pope starts -->
      <text>
<!-- biographical notice by editor -->
      </text>
      <text>
<!-- first poem -->
      </text>
      <text>
<!-- second poem -->
      </text>
     </group>
<!-- end of Pope section -->
<!-- ... -->
   </group>
<!-- end of 18th century section -->
   <group>
    <head>The Heyday</head>
<!-- texts and subgroups -->
   </group>
<!-- ... -->
  </group>
<!-- end of the anthology proper -->
  <back>
<!-- back matter for anthology -->
  </back>
</text>
```

Source: [76]

Note that the editor's introductory essays on each author may be treated as texts in their own right (as the essays on Lady Mary Wortley Montagu and Alexander Pope have been treated above), or as front matter to the embedded text, as the essay on Swift has been. The treatment in the example is intentionally inconsistent, to allow comparison of the two approaches. Consistency can be imposed either by treating the Swift section as a <group> containing one text by Swift and one by the editor, or by treating the Montagu and Pope sections as <text> elements containing the editor's essays as front matter. Marked in the second way, the Pope section of the book would look like this:

```
<text>
<!-- Section on Alexander Pope starts -->
  <front>
<!-- biographical notice by editor -->
  </front>
  <group>
    <text>
<!-- first poem -->
    </text>
    <text>
<!-- second poem -->
    </text>
  </group>
</text>
<!-- end of Pope section-->
```

The essays on 'The Eighteenth Century and the Grand Tour' and other larger sections could also be tagged as 'front' matter in the same way, by treating the larger sections as <text> elements rather than <group> elements.

Where, as in this case, an anthology contains different kinds of text (for example, mixtures of prose and drama, or transcribed speech and dictionary entries, or letters and verse), the elements to be encoded will of course be drawn from more than one module. The elements provided by the core module described in chapter *3. Elements Available in All TEI Documents* should however prove adequate for most simple purposes, where prose, drama, and verse are combined in a single collection.

For anthologies of short extracts such as commonplace books, it may often be preferable to regard each extract not as a text in its own right but simply as a quotation or <cit> element. The following component-level elements may be used to encode quotations of this kind:

<cit> (cited quotation) contains a quotation from some other document, together with a bibliographic reference to its source. In a dictionary it may contain an example text with at least one occurrence of the word form, used in the sense being described, or a translation of the headword, or an example.

<quote> (quotation) contains a phrase or passage attributed by the narrator or author to some agency external to the text.

For example, the chapter of 'extracts' which appears in the front matter of Melville's *Moby Dick* might be encoded as follows:

```
<div  n="2"  type="chap">
  <head>Extracts</head>
  <head>(Supplied by a sub-sub-Librarian)</head>
  <p>It will be seen that this mere painstaking burrower and
     grubworm of a poor devil of a Sub-Sub appears to have gone
     through the long Vaticans and street-stalls of the earth,
     picking up whatever random allusions to whales he could
     anyways find ...
     Here ye strike but splintered hearts together — there,
     ye shall strike unsplinterable glasses!</p>
  <p>
    <cit>
      <quote>And God created great whales.</quote>
      <bibl>Genesis</bibl>
    </cit>
    <cit>
      <quote>
        <l>Leviathan maketh a path to shine after him;</l>
        <l>One would think the deep to be hoary.</l>
      </quote>
      <bibl>Job</bibl>
    </cit>
    <cit>
      <quote>By art is created that great Leviathan,
            called a Commonwealth or State — (in Latin,
      <mentioned  xml:lang="la">civitas</mentioned>), which
            is but an artificial man.</quote>
      <bibl>Opening sentence of Hobbes's Leviathan</bibl>
    </cit>
  </p>
</div>
```

Source: [140]

For more information on the use of the <quote> and <bibl> elements, see sections *3.3.3. Quotation* and *3.11. Bibliographic Citations and References* respectively.

4.3.2 Floating Texts

An important characteristic of the unitary or composite text structures discussed so far is that they can be regarded as forming what is mathematically known as a tesselation covering the whole of the available text (or text division) at each hierarchic level. Just as an XML document has a single root element containing a single tree, each node of which forms a properly nested sub-tree, so it seems natural to think of the internal structure of a text as decomposable hierarchically into subparts, each of which is a properly nested subtree. While this is undoubtedly true of a large number of documents, it is not true of all. In particular, it is not true of texts which are only partly tesselated at a given level. For example, if a text A is contained by text B in such a way that part of B precedes A and part follows it, we cannot tesselate the whole of B. In such a case, we say that text A is a 'floating' text.

The <floatingText> element is a member of the model.divPart class, and can thus appear within any division level element in the same way as a paragraph. For example, texts such as the *Decameron* or the *Arabian Nights* might be regarded as containing many floating texts embedded within another single text, the framing narrative, rather than as groups of discrete texts in which the fragments of framing narrative are regarded as front or back matter.

As an example, we consider an 18th century text *The Lining to the **Patch-Work Screen***, by Jane Barker (1726). This lengthy narrative contains nearly a hundred distinct 'tales' embedded (as the title suggests) in a single patchwork. The work begins by introducing the central character, Galecia, but within a few pages launches into a distinct narrative, the story of Captain Manly:

```
<p>Galecia one Evening setting alone in her Chamber by a clear Fire,
and a clean Hearth ... reflected on the Providence of our
All-wise and Gracious Creator.... </p>
<p>She was thus ruminating, when a Gentleman enter'd the Room, the
Door being a jar... calling for a Candle, she beg'd a thousand
Pardons, engaged him to sit down, and let her know, what had so long
conceal'd him from her Correspondence.
</p>
<pb  n="5"/>
<floatingText>
  <body>
    <head>The Story of <hi>Captain Manly</hi>
    </head>
    <p>Dear Galecia, said he, though you partly know the loose, or rather
        lewd Life that I led in my Youth; yet I can't forbear relating part of
        it to you by way of Abhorrence...

<!-- Captain Manly's story here -->
        I had lost and spent all I had in the World; in which I verified the
        Old Proverb, That a Rolling Stone never gathers Moss,
    </p>
  </body>
</floatingText>
<pb  n="37"/>
```

Source: [9]

Following the conclusion of Captain Manly's tale, we are returned to Galecia, and almost immediately after that into two further stories. However, the Galecia narrative returns between each of the texts, which is why we choose to represent them as <floatingText>s:

```
<p>The Gentleman having finish'd his Story, Galecia waited on him to
the Stairs-head; and at her return, casting her Eyes on the Table, she
saw lying there an old dirty rumpled Book, and found in it the
following story: </p>
<floatingText>
  <body>
    <p> IN the time of the Holy War when
        Christians from all parts went into the Holy Land to oppose the Turks;
        Amongst these there was a certain English Knight...</p>
<!-- rest of story here -->
    <p>The King graciously pardoned the Knight; Richard was kindly receiv'd
        into his Convent, and all things went on in good order: But from hence
        came the Proverb, We must not strike <hi>Robert</hi> for
    <hi>Richard.</hi>
    </p>
  </body>
</floatingText>
<pb  n="43"/>
<p>By this time Galecia's Maid brought up her Supper; after which she
cast her Eyes again on the foresaid little Book, where she found the
following Story, which she read through before she went to bed.
</p>
<floatingText>
  <body>
    <head>The Cause of the Moors Overrunning
    <hi>Spain</hi>
    </head>
    <p>King ——— of Spain at his Death, committed the Government of his
        Kingdom to his Brother Don —— till his little Son should come of
        Age ...</p>
    <p>Thus the little Story ended, without telling what Misery
        befel the King and Kingdom, by the Moors, who over ran the Country for
        many Years after.  To which, we may well apply the Proverb,
    <quote>
       <l>Who drives the Devil's Stages,</l>
       <l>Deserves the Devil's Wages</l>
     </quote>
    </p>
  </body>
</floatingText>
<p>The reading this Trifle of a Story detained Galecia from her Rest
beyond her usual Hour; for she slept so sound the next Morning, that
she did not rise, till a Lady's Footman came to tell her, that his
Lady and another or two were coming to breakfast with her...
</p>
```

Source: [9]

In other multi-narrative texts, the individual nested tales may have greater significance than the framing narratives, and it may therefore be preferable to treat the fragments of framing narrative as front or back matter associated with each nested tale. This is commonly done, for example, in texts such as Chaucer's *Canterbury Tales*, where each tale is typically presented with front matter in which the teller of the tale is introduced, and back matter in which the pilgrims comment on it.

The <floatingText> element should only be used for complete texts which form a part of the text being encoded. Where a character in one narrative quotes from some other text or narrative, fully or in part, the <quote> element discussed in *3.3.3. Quotation* should be used instead.

4.4 Virtual Divisions

Where the whole of a division can be automatically generated, for example because it is derived from another part of this or another document, an encoder may prefer not to represent it explicitly but instead simply mark its location by means of a processing instruction, or by using the special purpose <divGen> element:

<divGen> (automatically generated text division) indicates the location at which a textual division generated automatically by a text-processing application is to appear.

This element is made available by the model.divGenLike class of which it is the sole element. The <divGen> element is a member of the att.typed class, from which it inherits the **type** and **subtype** attributes. It may appear wherever a or <div1> (<div2>, etc.) element may appear.

For example, if the table of contents (toc) for a given work is simply derived by copying the first <head> element from each element in a text, it might be more easily encoded as follows:

```
<divGen type="toc"/>
```

Similarly, in a digital edition combining a transcribed version of some text with a translated version of it, it may be desired to represent the transcript, the translation, and an aligned version of the two as three distinct divisions. This could be achieved by an encoding like the following:

```
<div>
<!-- transcript here-->
</div>
<div>
<!-- translation here -->
</div>
<divGen type="alignment"/>
```

The processing to be carried out when a <divGen> element is rendered will be determined by the application program or stylesheet in use: the function of the TEI markup is simply to identify the location at which the virtual division is to be generated, and also to provide some information about the kind of division to be generated. As such it may be regarded as a special kind of processing instruction, and could equally well be represented by one.

4.5 Front Matter

By front matter we mean distinct sections of a text (usually, but not necessarily, a printed one), prefixed to it by way of introduction or identification as a part of its production. Features such as title pages or prefaces are clear examples; a less definite case might be the prologue attached to a play. The front matter of an encoded text should not be confused with the TEI header described in chapter *2. The TEI Header*, which serves as a kind of front matter for the computer file itself, not the text it encodes.

An encoder may choose simply to ignore the front matter in a text, if the original presentation of the work is of no interest, or for other reasons; alternatively some or all components of the front matter may be thought worth including with the text as components of the <front> element.[1] With the exception of the title page, (on which see section *4.6. Title Pages*), front matter should be encoded using the same elements as the rest of a text. As with the divisions of the text body, no other specific tags are proposed here for the various kinds of subdivision which may appear within front matter: instead either numbered or un-numbered elements may be used. The following suggested values[2] for the **type** attribute may be used to distinguish various kinds of division characteristic of front matter:

[1] This decision should be recorded in the <samplingDecl> element of the header.

[2] As with all lists of 'suggested values' for attributes, it is recommended that software written to handle TEI-conformant texts be prepared to recognize and handle these values when they occur, without limiting the user to the values in this list.

preface A foreword or preface addressed to the reader in which the author or publisher explains the content, purpose, or origin of the text.

ack A formal declaration of acknowledgment by the author in which persons and institutions are thanked for their part in the creation of a text.

dedication A formal offering or dedication of a text to one or more persons or institutions by the author.

abstract A summary of the content of a text as continuous prose.

contents A table of contents, specifying the structure of a work and listing its constituents. The <list> element should be used to mark its structure.

frontispiece A pictorial frontispiece, possibly including some text.

The following extended example demonstrates how various parts of the front matter of a text may be encoded. The front part begins with a title page, which is presented in section *4.6. Title Pages* below. This is followed by a dedication and a preface, each of which is encoded as a distinct :

```
<div type="dedication">
  <p>To my parents, Ida and Max Fish</p>
</div>
<div type="preface">
  <head>Preface</head>
  <p>The answer this book gives to its title question is <q>there is
       and there isn't</q>.</p>
  <p>Chapters 1–12 have been previously published in the
     following journals and collections:
  <list>
     <item>chapters 1 and 3 in <title>New literary History</title>
     </item>
     <item>chapter 10 in <title>Boundary II</title> (1980)</item>
   </list>.
    I am grateful for permission to reprint.</p>
  <signed>S.F.</signed>
</div>
```

Source: [73]

The front matter concludes with another element, shown in the next example, this time containing a table of contents, which contains a <list> element (as described in section *3.7. Lists*). Note the use of the <ptr> element to provide page-references: the implication here is that the target identifiers supplied (fish1, fish2, etc.) will correspond with identifiers used for the elements containing chapters of the text itself. (For the <ptr> element, see *3.6. Simple Links and Cross-References.*)

```
<div type="contents">
  <head>Contents</head>
  <list>
    <item>Introduction, or How I stopped Worrying and Learned to Love
        Interpretation <ptr target="#fish1"/>
    </item>
    <item>
      <list>
        <head>Part One: Literature in the Reader</head>
```

```
      <item  n="1">Literature in the Reader: Affective Stylistics
      <ptr  target="#fish2"/>
      </item>
      <item  n="2">What is Stylistics and Why Are They Saying Such
             Terrible Things About It? <ptr  target="#fish3"/>
      </item>
    </list>
  </item>
 </list>
</div>
<div  xml:id="fish1">
 <head>Introduction</head>
<!-- .... -->
</div>
<div  xml:id="fish2">
 <head>Literature in the Reader</head>
<!-- .... -->
</div>
<div  xml:id="fish3">
 <head>What is stylistics?</head>
<!-- .... -->
</div>
```

Source: [73]

Alternatively, the pointers in the index might link to the page breaks at which a chapter begins, assuming that these have been included in the markup:

```
<!-- .... --><item  n="1">Literature in the Reader: Affective Stylistics
<ref  target="#fish-p24">24</ref>
</item>
<!-- .... -->
<div  type="chapter">
 <head>Literature in the Reader</head>
 <pb  xml:id="fish-p24"/>
<!-- .... -->
</div>
<!-- .... -->
```

The following example uses numbered divisions to mark up the front matter of a medieval text. Note that in this case no title page in the modern sense occurs; the title is simply given as a heading at the start of the front matter. Note also the use of the **type** attribute on the elements to indicate document elements comparatively unusual in modern books such as the initial prayer:

```
<front>
 <div1  type="incipit">
  <p>Here bygynniþ a book of contemplacyon, þe whiche
      is clepyd <title>þE CLOWDE OF VNKNOWYNG</title>,
      in þe whiche a soule is onyd wiþ GOD.</p>
 </div1>
 <div1  type="prayer">
  <head>Here biginneþ þe preyer on þe prologe.</head>
  <p>God, unto whom alle hertes ben open, & unto whome alle wille
      spekiþ, & unto whom no priue þing is hid: I beseche
```

```
      þee so for to clense þe entent of myn hert wiþ þe
      unspekable 3ift of þi grace, þat I may parfiteliche
      loue þee & worþilich preise þee. Amen.</p>
  </div1>
  <div1 type="preface">
    <head>Here biginneþ þe prolog.</head>
    <p>In þe name of þe Fader & of þe Sone &
      of þe Holy Goost.</p>
    <p>I charge þee & I beseeche þee, wiþ as moche
      power & vertewe as þe bonde of charite is sufficient
      to suffre, what-so-euer þou be þat þis book schalt
      haue in possession ...</p>
  </div1>
  <div1 type="contents">
    <head>Here biginneþ a table of þe chapitres.</head>
    <list>
      <label>þe first chapitre </label>
      <item>Of foure degrees of Cristen mens leuing; & of þe
        cours of his cleping þat þis book was maad vnto.</item>
      <label>þe secound chapitre</label>
      <item>A schort stering to meeknes & to þe werk of þis
        book</item>
      <label>þe fiue and seuenti chapitre</label>
      <item>Of somme certein tokenes bi þe whiche a man may proue
        whеþer he be clepid of God to worche in þis werk.</item>
    </list>
    <trailer>& here eendeþ þe table of þe chapitres.</trailer>
  </div1>
</front>
```

Source: [38]

If, however, the table of contents can be automatically generated from the remainder of the text, it may be preferable simply to mark its presence, either by means of an empty <divGen> element or by using an appropriate processing instruction.

4.6 Title Pages

Detailed analysis of the title page and other preliminaries of older printed books and manuscripts is of major importance in descriptive bibliography and the cataloguing of printed books; such analysis may require a rather more detailed module than that proposed here. The following elements are suggested as a means of encoding the major features of most title pages:

<titlePage> (title page) contains the title page of a text, appearing within the front or back matter.

<docTitle> (document title) contains the title of a document, including all its constituents, as given on a title page.

<titlePart> contains a subsection or division of the title of a work, as indicated on a title page.

@**type** specifies the role of this subdivision of the title.

<argument> A formal list or prose description of the topics addressed by a subdivision of a text.

<byline> contains the primary statement of responsibility given for a work on its title page or at the head or end of the work.

<docAuthor> (document author) contains the name of the author of the document, as given on the title page (often but not always contained in a byline).

<epigraph> contains a quotation, anonymous or attributed, appearing at the start of a section or chapter, or on a title page.

<imprimatur> contains a formal statement authorizing the publication of a work, sometimes required to appear on a title page or its verso.

<docEdition> (document edition) contains an edition statement as presented on a title page of a document.

<docImprint> (document imprint) contains the imprint statement (place and date of publication, publisher name), as given (usually) at the foot of a title page.

<docDate> (document date) contains the date of a document, as given (usually) on a title page.

<graphic/> indicates the location of an inline graphic, illustration, or figure.

Together with the <figure> element described in chapter *14. Tables, Formulæ, and Graphics*, these elements constitute the model.titlepagePart class. Any number of elements from this class can appear grouped together within a <titlePage> element. The <figure> element is included so as to enable encoders to record the presence of complex non-textual material on a title page. For simple cases such as printers' ornaments or illustrations the <graphic> element discussed in section *3.9. Graphics and other non-textual components* should be adequate.

The elements listed above, together with the <head> element, also constitute the class model.pLike.front. The elements in this class can appear within a minimal <front> element without any need to group them together and encode a complete title page.

Encoders wishing to add new elements to either class may do so using the methods described in section *23.2. Personalization and Customization*. Two examples of the use of these elements follow. First, the title page of the work discussed earlier in this section:

```
<front>
  <titlePage>
    <docTitle>
      <titlePart  type="main">Is There a Text in This Class?</titlePart>
      <titlePart  type="sub">The Authority of Interpretive Communities</titlePart>
    </docTitle>
    <docAuthor>Stanley Fish</docAuthor>
    <docImprint>
      <publisher>Harvard University Press</publisher>
      <pubPlace>Cambridge, Massachusetts</pubPlace>
      <pubPlace>London, England</pubPlace>
    </docImprint>
  </titlePage>
</front>
```

Source: [73]

Second, a characteristically verbose 17th century example. Note the use of the <lb> tag to mark the line breaks of the original where necessary:

```
<titlePage>
  <docTitle>
    <titlePart  type="main">THE
    <lb/>Pilgrim's Progress
    <lb/>FROM
    <lb/>THIS WORLD,
    <lb/>TO
    <lb/>That which is to come:</titlePart>
    <titlePart  type="sub">Delivered under the Similitude of a
```

```
    <lb/>DREAM</titlePart>
    <titlePart  type="desc">Wherein is Discovered,
    <lb/>The manner of his setting out,
    <lb/>His Dangerous Journey; And safe
    <lb/>Arrival at the Desired Countrey.</titlePart>
  </docTitle>
  <epigraph>
    <cit>
      <quote>I have used Similitudes,</quote>
      <bibl>Hos. 12.10</bibl>
    </cit>
  </epigraph>
  <byline>By <docAuthor>John Bunyan</docAuthor>.</byline>
  <imprimatur>Licensed and Entred according to Order.</imprimatur>
  <docImprint>
    <pubPlace>LONDON,</pubPlace>
     Printed for <name>Nath. Ponder</name>
    <lb/>at the <name>Peacock</name> in the <name>Poultrey</name>
    <lb/>near <name>Cornhil</name>, <docDate>1678</docDate>.
  </docImprint>
</titlePage>
```

Source: [22]

Where, as here, it is considered important to encode salient features of the way a title page was originally rendered, the techniques exemplified in *2.3.4. The Tagging Declaration* may also be useful.

Where title pages are encoded, their physical rendition is often of considerable importance. One approach to this requirement would be to use the <seg> tag, described in chapter *16. Linking, Segmentation, and Alignment*, to segment the typographic content of each part of the title page, and then use the global **rend** attribute to specify its rendition. Another would be to use a module specialized for the description of typographic entities such as pages, lines, rules, etc., bearing special-purpose attributes to describe line-height, leading, degree of kerning, font, etc. Further discussion of these problems is provided in chapter *11. Representation of Primary Sources*.

4.7 Back Matter

Conventions vary as to which elements are grouped as back matter and which as front. For example, some books place the table of contents at the front, and others at the back. Even title pages may appear at the back of a book as well as at the front. The content model for <back> and <front> elements are therefore identical.

The following suggested values may be used for the **type** attribute on all division elements, in order to distinguish various kinds of division characteristic of back matter:

appendix An ancillary self-contained section of a work, often providing additional but in some sense extra-canonical text.

glossary A list of terms associated with definition texts ('glosses'): this should be encoded as a **<list type="gloss">** (see section *3.7. Lists*).

notes A section in which textual or other kinds of notes are gathered together.

bibliogr A list of bibliographic citations: this should be encoded as a <listBibl> (see section *3.11. Bibliographic Citations and References*).

index Any form of index to the work.

colophon A statement appearing at the end of a book describing the conditions of its physical production.

No additional elements are proposed for the encoding of back matter at present. Some characteristic examples follow; first, an index (for the case in which a printed index is of sufficient interest to merit transcription):

```
<back>
  <div  type="index">
    <head>Index</head>
    <list  type="index">
      <item>Actors, public, paid for the contempt attending
            their profession, <ref>263</ref>
      </item>
      <item>Africa, cause assigned for the barbarous state of
            the interior parts of that continent, <ref>125</ref>
      </item>
      <item>Agriculture
      <list  type="indexentry">
         <item>ancient policy of Europe unfavourable to, <ref>371</ref>
         </item>
         <item>artificers necessary to carry it on, <ref>481</ref>
         </item>
         <item>cattle and tillage mutually improve each other, <ref>325</ref>
         </item>
         <item>wealth arising from more solid than that which proceeds
                  from commerce <ref>520</ref>
         </item>
       </list>
      </item>
      <item>Alehouses, not the efficient cause of drunkenness, <ref>461</ref>
      </item>
    </list>
  </div>
</back>
```

Source: [182]

Note that if the page breaks in the original source have also been explicitly encoded, and given identifiers, the references to them in the above index can more usefully be recorded as links. For example, assuming that the encoding of page 461 of the original source starts like this:

```
<pb  xml:id="P461"/>
```

then the last item above might be encoded more usefully in either of the following forms:

```
<item>Alehouses, not
the efficient cause of drunkenness, <ref  target="#P461">461</ref>
</item>
<item>Alehouses, not the efficient cause of drunkenness, <ptr  target="#P461"/>
</item>
```

Next, a back-matter division in epistolary form:

```
<back>
  <div  type="letter">
    <head>A letter written to his wife, founde with this booke
        after his death.</head>
    <p>The remembrance of the many wrongs offred thee, and thy
        unreproued vertues, adde greater sorrow to my miserable state,
        than I can utter or thou conceiue. ...
        ... yet trust I in the world to come to find mercie, by the
        merites of my Saiuour to whom I commend thee, and commit
        my soule.</p>
    <signed>Thy repentant husband for his disloyaltie,
    <name>Robert Greene.</name>
    </signed>
    <epigraph  xml:lang="la">
      <p>Faelicem fuisse infaustum</p>
    </epigraph>
    <trailer>FINIS</trailer>
  </div>
</back>
```

Source: [91]

And finally, a list of corrigenda and addenda with pseudo-epistolary features:

```
<back>
  <div  type="corrigenda">
    <head>Addenda</head>
    <salute  xml:lang="la">M. Scriblerus Lectori</salute>
    <p>Once more, gentle reader I appeal unto thee, from the shameful
        ignorance of the Editor, by whom Our own Specimen of
    <name>Virgil</name> hath been mangled in such miserable manner, that
        scarce without tears can we behold it. At the very entrance, Instead
        of <q  xml:lang="gr">προλεγομενα</q>, lo!
    <q  xml:lang="gr">προλεγωμενα</q> with an Omega!
        and in the same line <q  xml:lang="la">consulâs</q> with a circumflex!
        In the next page thou findest <q  xml:lang="la">leviter perlabere</q>,
        which his ignorance took to be the infinitive mood of
    <q  xml:lang="la">perlabor</q> but ought to be
    <q  xml:lang="la">perlabi</q> ... Wipe away all these
        monsters, Reader, with thy quill.</p>
  </div>
</back>
```

Source: [160]

4.8 Module for Default Text Structure

The module described by the present chapter has the following components:

Module textstructure: Default text structure

- Elements defined: TEI argument back body byline closer dateline div div1 div2 div3 div4 div5 div6 div7 docAuthor docDate docEdition docImprint docTitle epigraph floatingText front group imprimatur opener postscript salute signed text titlePage titlePart trailer

The selection and combination of modules to form a TEI schema is described in *1.2. Defining a TEI Schema*

Chapter 5

Representation of Non-standard Characters and Glyphs

Despite the availability of Unicode, text encoders still sometimes find that the published repertoire of available characters is inadequate to their needs. This is particularly the case when dealing with ancient languages, for which encoding standards do not yet exist, or where an encoder wishes to represent variant forms of a character or glyphs. The module defined by this chapter provides a mechanism to satisfy that need, while retaining compatibility with standards.

5.1 Is Your Journey Really Necessary?

When encoders encounter some graphical unit in a document which is to be represented electronically, the first issue to be resolved should be 'Is this really a different character?' To determine whether a particular graphical unit *is* a character or not, see *vi.2.2 Terminology and key concepts*.

If the unit is indeed determined to be a character, the next question should be 'Has this character been encoded already?' In order to determine whether a character has been encoded, encoders should follow the following steps:

1. Check the Unicode web site at `http://www.unicode.org`, in particular the page "Where is my Character?", and the associated character code charts. Alternatively, users can check the latest published version of *The Unicode Standard* (Unicode Consortium (2006)), though the web site is often more up to date than the printed version, and should be checked for preference.

   The pictures ('glyphs') in the Unicode code charts are only meant to be representative, not definitive. If a specific form of an already encoded character is required for a project, refer to the guidelines contained below under Annotating Characters. Remember that your encoded document may be rendered on a system which has different fonts from yours: if the specific form of a character is important to you, then you should document it.

2. Check the Proposed New Characters web page (`http://unicode.org/alloc/Pipeline.html`) to see whether the character is in line for approval.

3. Ask on the Unicode email list (`http://www.unicode.org/consortium/distlist.html`) to see whether a proposal is pending, or to determine whether this character is considered eligible for addition to the Unicode Standard.

Since there are now close to 100,000 characters in Unicode, chances are good that what you need is already there, but it might not be easy to find, since it might have a different name in Unicode. Look again, this

time at other sites, for example `http://www.eki.ee/letter`, which also provide searches based on scripts and languages. Take care, however, that all the properties of what seems to be a relevant character are consistent with those of the character you are looking for. For example, if your character is definitely a digit, but the properties of the best match you can find for it say that it is a letter, you may have a character not yet defined in Unicode.

In general, it is advisable to avoid Unicode characters generally described as presentation forms.[1] However, if the character you are looking for is being used in a notation (rather than as part of the orthography of a language) then it is quite acceptable to select characters from the Mathematical Operators block, provided that they have the appropriate properties (i.e. `So`: Symbol, Other; or `Sm`: Symbol, Math).

An encoded character may be precomposed or it may be formed from base characters and combining diacritical marks. Either will suffice for a character to be "found" as an encoded character.

If there are several possible Unicode characters to choose amongst, it is good practice to consult other colleagues and practitioners to see whether a consensus has emerged in favour of one or other of them.

If, however, no suitable form of your character seems to exist, the next question will be: 'Does the graphical unit in question represent a variant form of a known character, or does it represent a completely unencoded character?' If the character is determined to be missing from the Unicode Standard, it would be helpful to submit the new character for inclusion (see `http://unicode.org/pending/proposals.html`).

These guidelines will help you proceed once you have identified a given graphical unit as either a variant or an unencoded character. Determining this will require knowledge of the contents of the document that you have. The first case will be called *annotation* of a character, while the second case will be called *adding* of a new character. How to handle graphical units that represent variants will be discussed below (*5.3. Annotating Characters*) while the problem of representing new characters will be dealt with in section *5.4. Adding New Characters*.

While there is some overlap between these requirements, distinct specialized markup constructs have been created for each of these cases as explained in section *5.2. Markup Constructs for Representation of Characters and Glyphs* below. The following section will then proceed to discuss how to apply them to the problems at hand, discussing annotation of existing characters in section *5.3. Annotating Characters* and finally creation of new ones in *5.4. Adding New Characters*.

5.2 Markup Constructs for Representation of Characters and Glyphs

An XML document can, in principle, contain any defined Unicode character. The standard allows these characters to be represented either directly, using an appropriate encoding (UTF-8 by default), or indirectly by means of numeric character references (NCR), such as `Ä` (A-umlaut). The encoder can also restrict the range of characters which are represented directly in a document (or part of it) by adding a suitable encoding declaration. For example, if a document begins with the declaration `<?xml encoding="iso-8859-1"?>` any Unicode characters which are not in the ISO-8859-1 character set must be represented by NCRs.

The gaiji module defined by this chapter adds a further way of representing specific characters and glyphs in a document. This allows the encoder to distinguish characters and glyphs which Unicode regards as identical, to add new nonstandard characters or glyphs, and to represent Unicode characters not available in the document encoding by an alternative means.

The mechanism provided here consists functionally of two parts:

1. an element <g>, which serves as a proxy for new characters or glyphs
2. elements <char> and <glyph>, providing information about such characters or glyphs; these elements are stored in the <charDecl> element in the header.

[1] Specifically, characters in the Unicode blocks Alphabetic Presentation Forms, Arabic Presentation Forms-A, Arabic Presentation Forms-B, Letterlike Symbols, and Number Forms.

When the gaiji module is included in a schema, the <charDecl> element is added to the model.encodingPart class, and the <g> element is added to the phrase class. These elements and their components are documented in the rest of this section.

The Unicode standard defines properties for all the characters it defines in the Unicode Character Database, knowledge of which is usually built into text processing systems. If the character represented by the <g> element does not exist in Unicode at all, its properties are not available. If the character represented is an existing Unicode character, but is not available in the document character set recognized by a given text processing system, it may also be convenient to have access to its properties in the same way. The <char> element makes it possible to store properties for use by such applications in a standard way.

The list of attributes (properties) for characters is modelled on those in the Unicode Character Database, which distinguishes normative and informative character properties. Additional, non-Unicode, properties may also be supplied. Since the list of properties will vary with different versions of the Unicode Standard, there may not be an exact correspondence between them and the list of properties defined in these Guidelines.

Usage examples for these elements are given below at *5.3. Annotating Characters* and *5.4. Adding New Characters*. The gaiji module itself is formally defined in section *5.6. Module Character and Glyph Documentation* below. It declares the following additional elements:

<charDecl> (character declarations) provides information about nonstandard characters and glyphs.

<g> (character or glyph) represents a non-standard character or glyph.

@**ref** points to a description of the character or glyph intended.

The <charDecl> element is a member of the class model.encodingPart, and thus becomes available within <encodingDesc> when this module is included in a schema. The <g> element is the only member of the class model.gLike: this class is referenced as an alternative to plain text in almost every element which contains plain text, thus permitting the <g> element also to appear at such places when this module is included in a schema.

The following elements may appear within a <charDecl> element:

<desc> (description) contains a brief description of the intended usage, purpose, or application of its parent element.

<char> (character) provides descriptive information about a character.

<glyph> (character glyph) provides descriptive information about a character glyph.

The <char> and <glyph> elements have similar contents and are used in similar ways, but their functions are different. The <char> element is provided to define a character which is not available in the current document character set, for whatever reason, as stated above. The <glyph> element is used to annotate a character that has already been defined somewhere (either in the document character set, or through a <char> element) by providing a specific glyph that shows how a character appeared in the original document. This is necessary since Unicode code points refer not to a single, specific glyph shape of a character, but rather to a set of glyphs, any of which may be used to render the code point in question; in some cases they can differ considerably.

The <glyph> element is provided for cases where the encoder wants to specify a specific glyph (or family of glyphs) out of all possible glyphs. Unfortunately, due to the way Unicode has been defined, there are cases where several glyphs that logically belong together have been given separate code points, especially in the blocks defining East Asian characters. In such cases, <glyph> elements can also be used to express the view that these apparently distinct characters are to be regarded as instances of the same character (see further *5.3. Annotating Characters*).

The Unicode Standard recommends naming conventions which should be followed strictly where the intention is to annotate an existing Unicode character, and which may also be used as a model when creating

new names for characters or glyphs[2]. For convenience of processing, the following distinct elements are proposed for naming characters and glyphs:

<charName> (character name) contains the name of a character, expressed following Unicode conventions.

<glyphName> (character glyph name) contains the name of a glyph, expressed following Unicode conventions for character names.

Within both <char> and <glyph>, the following elements are available:

<gloss> identifies a phrase or word used to provide a gloss or definition for some other word or phrase.

<charProp> (character property) provides a name and value for some property of the parent character or glyph.

<desc> (description) contains a brief description of the intended usage, purpose, or application of its parent element.

<mapping> (character mapping) contains one or more characters which are related to the parent character or glyph in some respect, as specified by the **type** attribute.

<graphic/> indicates the location of an inline graphic, illustration, or figure.

Four of these elements (<gloss>, <desc>, <graphic>, and <remarks>) are defined by other TEI modules, and their usage here is no different from their usage elsewhere. The <graphic> element, however, is used here only to link to an image of the character or glyph under discussion, or to contain a representation of it in SVG. Several <graphic> elements may be given, for example to provide images with different resolution, or in different formats. The **mimeType** attribute which <graphic> acquires from its membership of the att.internetMedia class may be used to specify the format of the image.

The <mapping> element is similar to the standard TEI <equiv> element. While the latter is used to express correspondence relationships between TEI concepts or elements and those in other systems or ontologies, the former is used to express any kind of relationship between the character or glyph under discussion and characters or glyphs defined elsewhere. It may contain any Unicode character, or a <g> element linked to some other <char> or <glyph> element, if, for example, the intention is to express an association between two non-standard characters. The type of association is indicated by the **type** attribute, which may take such values as `exact` for exact equivalences, `uppercase` for uppercase equivalences, `lowercase` for lowercase equivalences, `standardized` for standardized forms, and `simplified` for simplified characters, etc., as in the following example:

```
<charDecl>
  <char  xml:id="aenl">
    <charName>LATIN LETTER ENLARGED SMALL A</charName>
    <charProp>
      <localName>entity</localName>
      <value>aenl</value>
    </charProp>
    <mapping  type="standardized">a</mapping>
  </char>
</charDecl>
```

The mapping element may also be used to represent a mapping of the character or (more likely) glyph under discussion onto a character from the private use area as in this example:

[2] It should be noted, however, that this naming convention cannot meaningfully be applied to East Asian characters; the typical Unicode descriptions for these characters take the form 'CJK Unified Ideograph `U+4E00`', where `U+4E00` is simply the Unicode code point value of the character in question. In cases where no Unicode code point exists, there is little hope of finding a name that helps to identify the character. Names should therefore be constructed in a way meaningful to local practice, for example by using a reference number from a well-known character dictionary or a project-specific serial number.

```
<charDecl>
  <glyph  xml:id="z103">
    <glyphName>LATIN LETTER Z WITH TWO STROKES</glyphName>
    <mapping  type="standardized">Z</mapping>
    <mapping  type="PUA">U+E304</mapping>
  </glyph>
</charDecl>
```

A more precise documentation of the properties of any character or glyph may be supplied using the generic <charProp> element described in the next section. Despite its name, this element may be used for either characters or glyphs.

5.2.1 Character Properties

The Unicode Standard documents 'ideal' characters, defined by reference to a number of properties (or attribute-value pairs) which they are said to possess. For example, a lowercase letter is said to have the value `Ll` for the property `general-category`. The Standard distinguishes between normative properties (i.e. properties which form part of the definition of a given character), and informative or additional properties which are not normative. It also allows for the addition of new properties, and (in some circumstances) alteration of the values currently assigned to certain properties. When making such modifications, great care should be taken not to override standard informative properties for characters which already exist in the Unicode Standard, as documented in Freytag (2006).

The <charProp> element allows an encoder to supply information about a character or glyph. Where the information concerned relates to a property which has already been identified in the Unicode Standard, encoders are urged to use the appropriate Unicode property name.

The following elements are used to record character properties:

<unicodeName> (unicode property name) contains the name of a registered Unicode normative or informative property.

<localName> (locally-defined property name) contains a locally defined name for some property.

<value> (value) contains a single value for some property, attribute, or other analysis.

For each property, the encoder must supply either a <unicodeName> or a <localName>, followed by a <value>.

For convenience, we list here some of the normative character properties and their values. For full information, refer to chapter 4 of *The Unicode Standard*, or the online documentation of the Unicode Character Database.

general-category The general category (described in the Unicode Standard chapter 4 section 5) is an assignment to some major classes and subclasses of characters. Suggested values for this property are listed here:

Lu Letter, uppercase

Ll Letter, lowercase

Lt Letter, titlecase

Lm Letter, modifier

Lo Letter, other

Mn Mark, nonspacing

Mc Mark, spacing combining

Me Mark, enclosing

Nd Number, decimal digit

Nl Number, letter

No Number, other

Pc Punctuation, connector

Pd Punctuation, dash

Ps Punctuation, open

Pe Punctuation, close

Pi Punctuation, initial quote

Pf Punctuation, final quote

Po Punctuation, other

Sm Symbol, math

Sc Symbol, currency

Sk Symbol, modifier

So Symbol, other

Zs Separator, space

Zl Separator, line

Zp Separator, paragraph

Cc Other, control

Cf Other, format

Cs Other, surrogate

Co Other, private use

Cn Other, not assigned

directional-category This property applies to all Unicode characters. It governs the application of the algorithm for bi-directional behaviour, as further specified in Unicode Annex 9, *The Bidirectional Algorithm*. The following 19 different values are currently defined for this property in Davis et al (2006):

L left to right

LRE left to right embedding

LRO left to right override

R right to left

AL right to left Arabic

RLE right to left embedding

RLO right to left override

PDF Pop Directional Format

EN European Number

ES European Number Separator

ET European Number Terminator

AN Arabic Number

CS Common Number Separator

NSM Non-spacing Mark

BN Boundary Neutral

B Paragraph separator

S Segment separator

WS Whitespace

ON Other neutrals

canonical-combining-class This property exists for characters that are not used independently, but in combination with other characters, for example the strokes making up CJK (Chinese, Japanese, and Korean) characters. It records a class for these characters, which is used to determine how they interact typographically. The following values are defined in the Unicode Standard 5.0: (see Unicode Character Database: Canonical Combining Class Values)

0 Spacing, split, enclosing, reordrant, and Tibetan subjoined

1 Overlays and interior

7 Nuktas

8 Hiragana/Katakana voicing marks

9 Viramas

10 Start of fixed position classes

199 End of fixed position classes

200 Below left attached

202 Below attached

204 Below right attached

208 Left attached (reordrant around single base character)

210 Right attached

212 Above left attached

214 Above attached

216 Above right attached

218 Below left

220 Below

222 Below right

224 Left (reordrant around single base character)

226 Right

228 Above left

230 Above

232 Above right

233 Double below

234 Double above

240 Below (iota subscript)

character-decomposition-mapping This property is defined for characters, which may be decomposed, for example to a canonical form plus a typographic variation of some kind. For such characters the Unicode standard specifies both a decomposition type and a decomposition mapping (i.e. another Unicode character to which this one may be mapped in the way specified by the decomposition type). The following types of mapping are defined in the Unicode Standard:

font A font variant (e.g. a blackletter form)

noBreak A no-break version of a space or hyphen

initial An initial presentation form (Arabic)

medial A medial presentation form (Arabic)

final A final presentation form (Arabic)

isolated An isolated presentation form (Arabic)

circle An encircled form

super A superscript form

sub A subscript form

vertical A vertical layout presentation form

wide A wide (or zenkaku) compatibility character

narrow A narrow (or hankaku) compatibility character

small A small variant form (CNS compatibility)

square A CJK squared font variant

fraction A vulgar fraction form

compat Otherwise-unspecified compatibility character

numeric-value This property applies for any character which expresses any kind of numeric value. Its value is the intended value in decimal notation.

mirrored The mirrored character property is used to properly render characters such as U+0028, `OPENING PARENTHESIS` independent of the text direction: it has the value `Y` (character is mirrored) or `N` (code is not mirrored).

The Unicode Standard also defines a set of informative (but non-normative) properties for Unicode characters. If encoders want to provide such properties, they may be included using the suggested Unicode name, tagged using the <unicodeName> element. However, encoders may also supply other locally-defined properties, which must be named using the <localName> element to distinguish them. If a Unicode name exists for a given property, it should however always be preferred to a locally defined name. Locally defined names should be used only for properties which are not specified by the Unicode Standard.

5.3 Annotating Characters

Annotation of a character becomes necessary when it is desired to distinguish it on the basis of certain aspects (typically, its graphical appearance) only. In a manuscript, for example, where distinctly different forms of the letter "r" can be recognized, it might be useful to distinguish them for analytic purposes, quite distinct from the need to provide a accurate representation of the page. A digital facsimile, particularly one linked

to a transcribed and encoded version of the text, will always provide a superior visual representation (for information on how to link a digital facsimile to a transcribed text see *11.1. Digital Facsimiles*), but cannot be used to support arguments based on the distribution of such different forms. Character annotation as described here provides a solution to this problem.[3]

Assuming that we wish to distinguish the variant glyphs from the standard representation for the character concerned, we will need to define distinct <glyph> elements, one for each of the forms of the letter we wish to distinguish:

```
<charDecl>
  <glyph xml:id="r1">
    <glyphName>LATIN SMALL LETTER R WITH ONE FUNNY STROKE</glyphName>
    <charProp>
      <localName>entity</localName>
      <value>r1</value>
    </charProp>
    <graphic url="r1img.png"/>
  </glyph>
  <glyph xml:id="r2">
    <glyphName>LATIN SMALL LETTER R WITH TWO FUNNY STROKES</glyphName>
    <charProp>
      <localName>entity</localName>
      <value>r2</value>
    </charProp>
    <graphic url="r2img.png"/>
  </glyph>
</charDecl>
```

With these definitions in place, occurrences of these two special "r"s in the text can be annotated using the element <g>:

```
<p>Wo<g ref="#r1">r</g>ds in this
manusc<g ref="#r2">r</g>ipt are sometimes
written in a funny way.</p>
```

As can be seen in this example, the <glyph> element pointed to from the <g> element will be interpreted as an annotation on the content of the element <g>. This mechanism can also be used to indicate ligatures, as in the following example:

```
<p> ... <g ref="#Filig">Fi</g>lthy riches...</p>
<!-- in the charDecl -->
<glyph xml:id="Filig">
  <glyphName>LATIN UPPER F AND LATIN LOWER I LIGATURE</glyphName>
  <graphic url="Filig.png"/>
</glyph>
```

(In fact the Unicode Standard does provide a character to represent the `Fi` ligature; the encoder may however prefer not to use it in order to simplify other text processing operations, such as indexing).

With this markup in place, it will be possible to write programs to analyze the distribution of the different letters "r" as well as produce more 'faithful' renderings of the original. It will also be possible to produce normalized versions by simply ignoring the annotation pointed to by the element <g>.

[3] It should be kept in mind that any kind of text encoding is an abstraction and an interpretation of the text at hand, which will not necessarily be useful in reproducing an exact facsimile of the appearance of a manuscript.

For brevity of encoding, it may be preferred to predefine internal entities such as the following:

```
<!ENTITY r1 '<g ref="#r1">r</g>' >
<!ENTITY r2 '<g ref="#r2">r</g>' >
```

which would enable the same material to be encoded as follows:

```
<p>Wo&r1;ds in this manusc&r2;ipt are
sometimes written in a funny way.</p>
```

The same technique may be used to represent particular abbreviation marks as well as to represent other characters or glyphs. For example, if we believe that the r-with-one-funny-stroke is being used as an abbreviation for `receipt`, this might be represented as follows:

```
<abbr>&r1;</abbr>
```

Note however that this technique employs markup objects to provide a link between a character in the document and some annotation on that character. Therefore, it cannot be used in places where such markup constructs are not allowed, notably in attribute values.

Since the need to use these constructs to annotate or define characters occurs frequently in Chinese, Korean, and Japanese documents, here are some issues that are specific to these documents. There are two slightly different versions of the problem. In the first case, due to the way Unicode is defined, there are occasions when more than one glyph is defined for a character. In such an occasion, one might want to retain the character as used, but add information in a way so that a normalizer (for search or indexing operations) could take advantage of this information. To achieve this, we simply define within a <charDecl> element a <glyph> that has two <mapping> elements, as shown here:

```
<charDecl>
  <glyph  xml:id="u8aaa">
    <mapping  type="Unicode">説</mapping>
    <mapping  type="orthographic">説</mapping>
  </glyph>
</charDecl>
```

The first of these <mapping>s, of type **Unicode**, simply maps our glyph to the code point where Unicode defined it. The other one, of type **Standard**, encodes the fact that in our view, this glyph is a variation of the standard character given in the content of the element. We could then use this <glyph> element's unique identifier **u8aaa** to refer to it from within a text as follows.

```
<g  ref="#u8aaa">説</g>
```

A slightly different, but related problem occurs when we have multiple variants, none of which has been defined in Unicode. In this case, we need to define one as a new character using <char>, and the others as glyphs using <glyph>.

```
<charDecl>
  <char xml:id="newchar1">
<!-- more properties here -->
  </char>
  <glyph xml:id="varofnewchar1">
<!-- more properties here -->
    <mapping type="Standard">
      <g ref="#newchar1"/>
    </mapping>
  </glyph>
</charDecl>
```

The <char> defines a new character, while the <glyph> element then defines a variant glyph of this newly defined character. Additional properties should be specified in order to make these both identifiable.

5.4 Adding New Characters

The creation of additional characters for use in text encoding is quite similar to the annotation of existing characters. The same element <g> is used to provide a link from the character instance in the text to a character definition provided within the <charDecl> element. This character definition takes the form of a <char> element. The element <g> itself will usually be empty, but could contain a code point from the Private Use Area (PUA) of the Unicode Standard, which is an area set aside for the very purpose of privately adding new characters to a document. Recommendations on how to use such PUA characters are given in the following section.

In some circumstances, it may be desirable to provide a single precomposed form of a character that is encoded in Unicode only as a sequence of code points. For example, in Medieval Nordic material, a character looking like a lowercase letter Y with a dot and an acute-accent above it may be encountered so frequently that the encoder wishes to treat it as a single precomposed character with one single coded value. In the transcription concerned, the encoder enters this letter as `&ydotacute;`, which when the transcription is processed can then be expanded in one of three ways, depending on the mapping in force. The entity reference might be translated into the sequence of corresponding Unicode code points or into some locally-defined PUA character (say ``) for local processing only. Both these options have disadvantages; the former loses the fact that the sequence of composed characters is regarded as a single object; the second is not reliably portable. Therefore, the recommended representation is to use the <g> element defined by the module defined in this chapter:

```
<g ref="#ydotacute"/>
```

. This makes it possible for the encoder to provide useful documentation for the particular character or glyph so referenced:

```
<char xml:id="ydotacute">
  <charName>LATIN SMALL LETTER Y WITH DOT ABOVE AND
    ACUTE</charName>
  <charProp>
    <localName>entity</localName>
    <value>ydotacute</value>
  </charProp>
  <mapping type="composed">&#x0079;&#x0307;&#x0301;</mapping>
  <mapping type="PUA">U+E0A4</mapping>
</char>
```

This definition specifies the mapping between this composed character and the individual Unicode-defined code points which make it up. It also supplies a single locally-defined property ('entity') for the character concerned, the purpose of which is to supply a recommended character entity name for the character.

Under certain circumstances, Chinese Han characters can be written within a circle. Rather than considering this as simply an aspect of the rendering, an encoder may wish to treat such circled characters as entirely distinct derived characters. For a given character (say that represented by the numeric-character reference `人`) the circled variant might conveniently be represented as

```
<g ref="#U4EBA-circled"/>
```

, which references a definition such as the following:

```
<char xml:id="U4EBA-circled">
  <charName>CIRCLED IDEOGRAPH</charName>
  <charProp>
    <unicodeName>character-decomposition-mapping</unicodeName>
    <value>circle</value>
  </charProp>
  <charProp>
    <localName>daikanwa</localName>
    <value>36</value>
  </charProp>
  <mapping type="standard"> &#x4EBA;
  </mapping>
  <mapping type="PUA"> &#xE000;
  </mapping>
</char>
```

In this example, the 'circled ideograph' character has been defined with two mappings, and with two properties. The two properties are the Unicode-defined character-decomposition which specifies that this is a circled character, using the appropriate terminology (see *5.2.1. Character Properties* above) and a locally defined property known as 'daikanwa'. The two mappings indicate firstly that the standard form of this character is the character `人`, and secondly that the character used to represent this character locally is the PUA character ``. For convenience of local processing this PUA character may in fact appear as content of the <g> element. In general, however, the <g> element will be empty.

5.5 How to Use Code Points from the Private Use Area

The developers of the Unicode Standard have set aside an area of the codespace for the private use of software vendors, user groups, or individuals. As of this writing (Unicode 5.0), there are around 137,000 code points available in this area, which should be enough for most needs. No code point assignments will be made to this area by standard bodies and only some very basic default properties have been assigned (which may be overwritten where necessary by the mechanism outlined in this chapter). Therefore, unlike all other code points defined by the Unicode Standard, PUA code points should *not* be used directly in documents intended for blind interchange.

In the two previous examples, we mentioned that the variant characters concerned might well be assigned specific code points from the PUA. This might, for example, facilitate the use of a particular font which displays the desired character at this code point in the local processing environment. Since however this assignment would be valid only on the local site, documents containing such code points are unsuitable for blind interchange. During the process of preparing such documents for interchange, any PUA code points should be replaced by an appropriate use of the <g> element, such as **<g ref="#xxxx">**, thus associating the

character required with the documentation of it provided by the referenced <char> element. The PUA character used during the preparation of the document might be recorded in the <char> element, as shown in the example in *5.4. Adding New Characters*, or retained as content of the <g> element. However, since there is no requirement that the same PUA character be used to represent it at the receiving site, and since it may well be the case that this other site has already made an assignment of some other character to the original PUA code point, it is best practice to remove the locally-defined PUA character. It is to be expected that a further translation into the local processing environment at the receiving site will be necessary to handle such characters, during which variant letters can be converted to hitherto unused code points on the basis of the information provided in the <char> element.

This mechanism is rather weak in cases where DOM trees or parsed XML fragments are exchanged, which may increasingly be the case. The best an application can do here is to treat any occurrence of a PUA character only in the context of the local document and use the properties provided through the <char> element as a handle to the character in other contexts.

In the fullness of time, a character may become standardized, and thus assigned a specific code point outside the PUA. Documents which have been encoded using the mechanism must at the least ensure that this changed code point is recorded within the relevant <char> element; it will however normally be simpler to remove the <char> element and replace all occurrences of <g> elements which reference it by occurrences of the newly coded character.

5.6 Module Character and Glyph Documentation

The module described in this chapter makes available the following components:

Module gaiji: Character and glyph documentation

- Elements defined: char charDecl charName charProp g glyph glyphName localName mapping unicodeName value

The selection and combination of modules to form a TEI schema is described in *1.2. Defining a TEI Schema*.

Chapter 6

Verse

This module is intended for use when encoding texts which are entirely or predominantly in verse, and for which the elements for encoding verse structure already provided by the core module are inadequate.

The tags described in section *3.12.1. Core Tags for Verse* include elements for the encoding of verse lines and line groups such as stanzas: these are available for any TEI document, irrespective of the module it uses. Like the modules for prose and for drama, the module for verse additionally makes use of the module defined in chapter *4. Default Text Structure* to define the basic formal structure of a text, in terms of <front>, <body> and <back> elements and the text-division elements into which these may be subdivided.

The module for verse extends the facilities provided by these modules in the following ways:

- a special purpose <caesura> element is provided, to allow for segmentation of the verse line (see section *6.2. Components of the Verse Line*)
- a set of attributes is provided for the encoding of rhyme scheme and metrical information (see sections *6.3. Rhyme and Metrical Analysis* and *6.4. Rhyme*)
- a special purpose <rhyme> element is provided to support simple analysis of rhyming words (see section *6.4. Rhyme*)

6.1 Structural Divisions of Verse Texts

Like other kinds of text, texts written in verse may be of widely differing lengths and structures. A complete poem, no matter how short, may be treated as a free-standing text, and encoded in the same way as a distinct prose text. A group of poems functioning as a single unit may be encoded either as a <group> or as a <text>, depending on the encoder's view of the text. For further discussion, including an example encoding for a verse anthology, see chapter *4. Default Text Structure*.

Many poems consist only of ungrouped lines. This short poem by Emily Dickinson is a simple case:

```
<text>
  <front>
    <head>1755</head>
  </front>
  <body>
    <l>To make a prairie it takes a clover and one bee,</l>
    <l>One clover, and a bee,</l>
    <l>And revery.</l>
    <l>The revery alone will do,</l>
    <l>If bees are few.</l>
  </body>
</text>
```

Source: [60]

Often, however, lines are grouped, formally or informally, into stanzas, verse paragraphs, etc. The <lg> element defined in the core tag set (in section *3.12.1. Core Tags for Verse*) may be used for all such groupings. It may thus serve for informal groupings of lines such as those of the following example from Allen Ginsberg:

```
<text>
  <body>
    <head>My Alba</head>
    <lg>
      <l>Now that I've wasted</l>
      <l>five years in Manhattan</l>
      <l>life decaying</l>
      <l>talent a blank</l>
    </lg>
    <lg>
      <l>talking disconnected</l>
      <l>patient and mental</l>
      <l>sliderule and number</l>
      <l>machine on a desk</l>
    </lg>
  </body>
</text>
```

Source: [87]

It may also be used to mark the verse paragraphs into which longer poems are often divided, as in the following example from Samuel Taylor Coleridge's *Frost at Midnight*:

```
<lg>
  <l>The Frost performs its secret ministry,</l>
  <l>Unhelped by any wind. ...</l>
  <l>Whose puny flaps and freaks the idling Spirit</l>
  <l>By its own moods interprets, every where</l>
  <l>Echo or mirror seeking of itself,</l>
  <l  part="I">And makes a toy of Thought.</l>
</lg>
<lg>
  <l  part="F">But O! how oft,</l>
  <l>How oft, at school, with most believing mind</l>
  <l>Presageful, have I gazed upon the bars,</l>
  <l>To watch that fluttering <hi>stranger</hi>! ... </l>
</lg>
<lg>
  <l>Dear Babe, that sleepest cradled by my side,</l>
</lg>
```

Source: [43]

Note, in the above example, the use of the **part** attribute on the <l> element, where a verse line is broken between two line groups, as discussed in section *3.12.1. Core Tags for Verse*.

Most typically, however, the <lg> element is used to mark the highly regular line groups which characterize stanzaic and similar verse forms, as in the following example from Chaucer:

```
<lg>
  <l>Sire Thopas was a doghty swayn;</l>
  <l>White was his face as payndemayn,</l>
  <l>His lippes rede as rose;</l>
  <l>His rode is lyk scarlet in grayn,</l>
  <l>And I yow telle in good certayn,</l>
  <l>He hadde a semely nose.</l>
</lg>
<lg>
  <l>His heer, his ber was lyk saffroun,</l>
  <l>That to his girdel raughte adoun;</l>
</lg>
```

Source: [35]

Like other text-division elements, <lg> elements may be nested hierarchically. For example, one particularly common English stanzaic form consists of a quatrain or sestet followed by a couplet. The <lg> element may be used to encode both the stanza and its components, as in the following example from Byron:

```
<lg  type="stanza">
  <lg  type="sestet">
    <l>In the first year of Freedom's second dawn</l>
    <l>Died George the Third; although no tyrant, one</l>
    <l>Who shielded tyrants, till each sense withdrawn</l>
    <l>Left him nor mental nor external sun:</l>
    <l>A better farmer ne'er brushed dew from lawn,</l>
    <l>A worse king never left a realm undone!</l>
  </lg>
  <lg  type="couplet">
    <l>He died — but left his subjects still behind,</l>
    <l>One half as mad — and t'other no less blind.</l>
  </lg>
</lg>
```

Source: [27]

Note the use of the **type** attribute to name the type of unit encoded by the <lg> element; this attribute is common to all members of the att.divLike class (see section *4.1.1. Un-numbered Divisions*).[1] 'Sestet' and 'couplet' might conceivably also be used as the values of the **rhyme** attribute in an analysis of rhyme scheme, for which see below, section *6.3. Rhyme and Metrical Analysis*. The **type** attribute is intended solely for conventional names of different classes of text block; the **met** attribute is intended for systematic metrical analysis.

As a further example, consider the Shakespearean sonnet. This may be divided into two parts: a concluding couplet, and a body of twelve lines, itself subdivided into three quatrains:

```
<text>
  <body>
    <lg>
      <lg  type="quatrain">
        <l>My Mistres eyes are nothing like the Sunne,</l>
        <l>Currall is farre more red, then her lips red</l>
        <l>If snow be white, why then her brests are dun:</l>
        <l>If haires be wiers, black wiers grown on her head:</l>
```

[1] For discussion of other attributes of this class, see *4.1.4. Partial and Composite Divisions*.

```
    </lg>
    <lg  type="quatrain">
      <l>I have seene Roses damaskt, red and white,</l>
      <l>But no such Roses see I in her cheekes,</l>
      <l>And in some perfumes is there more delight,</l>
      <l>Then in the breath that from my Mistres reekes.</l>
    </lg>
    <lg  type="quatrain">
      <l>I love to heare her speake, yet well I know,</l>
      <l>That Musicke hath a farre more pleasing sound:</l>
      <l>I graunt I never saw a goddesse goe,</l>
      <l>My Mistres when shee walkes treads on the ground.</l>
    </lg>
  </lg>
  <lg  type="couplet">
    <l>And yet by heaven I think my love as rare,</l>
    <l>As any she beli'd with false compare.</l>
  </lg>
 </body>
</text>
```

Source: [178]

Particularly lengthy poetic texts are often subdivided into units larger than stanzas or paragraphs, which may themselves be subdivided. Spenser's *Faery Queene*, for example, consists of twelve 'books' each of which contains a prologue followed by twelve 'cantos'. Each prologue and each canto consists of nine-line 'stanzas', each of which follows the same regular pattern. Other examples in the same tradition are easy to find.

Large structures of this kind are most conveniently represented by or <div1> elements, as described in section *4.1. Divisions of the Body*. Thus the start of the *Faerie Queene* might be encoded as follows:

```
<body>
 <div  n="I"  type="book">
  <div  n="1"  type="canto">
   <lg  n="I.1.1"  type="stanza">
    <l>A Gentle Knight was pricking on the plain</l>
    <l>Y cladd in mightie armes and silver shielde,</l>
   </lg>
  </div>
 </div>
</body>
```

Source: [186]

The encoder must choose at which point in the hierarchy of structural units to introduce <lg> elements rather than a yet smaller element: it would (for example) also be possible to encode the above example as follows:

```
<body>
 <div  n="I"  type="book">
  <div  n="I.1"  type="canto">
   <div  n="I.1.1"  type="stanza">
    <l>A noble knight was pricking on the plain</l>
    <l>Ycladd in mightie armes and silver shielde,</l>
   </div>
```

```
   </div>
  </div>
</body>
```

One reason for using rather than <lg> elements is that the former may contain non-metrical elements, such as epigraphs or dedications and other members of the model.divTop class, whereas <lg> elements may contain only headings or metrical lines.

6.2 Components of the Verse Line

It is often convenient for various kinds of analysis to encode subdivisions of verse lines. The general purpose <seg> element defined in the tag set for segmentation and alignment (section *16.3. Blocks, Segments, and Anchors*) is provided for this purpose:

<seg> (arbitrary segment) represents any segmentation of text below the 'chunk' level.

To use this element together with the module for verse, the module for segmentation and alignment must also be enabled as further described in section *1.2. Defining a TEI Schema*.

In Old and Middle English alliterative verse, individual verse lines are typically split into half lines. The <seg> element may be used to mark these explicitly, as in the following example from Langland's *Piers Plowman*:

```
<l>
  <seg>In a somer seson,</seg>
  <seg>whan softe was the sonne,</seg>
</l>
<l>
  <seg>I shoop me into shroudes</seg>
  <seg>as I a sheep were,</seg>
</l>
<l>
  <seg>In habite as an heremite </seg>
  <seg>unholy of werkes,</seg>
</l>
<l>
  <seg>Went wide in this world </seg>
  <seg>wondres to here.</seg>
</l>
```

Source: [118]

The <seg> element can be nested hierarchically, in the same way as the <lg> element, down to whatever level of detailed structure is required. In the following example, the line has been divided into feet, each of which has been further subdivided into syllables.[2]

```
<l>
  <seg  type="foot">
    <seg  type="syll">Ar</seg>
    <seg  type="syll">ma </seg>
    <seg  type="syll">vi</seg>
  </seg>
```

[2] As elsewhere in these Guidelines, this example has been formatted for clarity of exposition rather than correct display. Note in particular that whether an XML processor retains whitespace within the <seg> element or not (this can be configured by means of the **xml:space** attribute) this example will still require additional processing, since white space should be retained for the lower level <seg> elements (those of type **syll**) but not for the higher level one (those of type **foot**).

```
  <seg type="foot">
    <seg type="syll">rum</seg>
    <seg type="syll">que </seg>
    <seg type="syll">ca</seg>
  </seg>
  <seg type="foot">
    <seg type="syll">no </seg>
    <seg type="syll">Tro</seg>
  </seg>
  <seg type="foot">
    <seg type="syll">iae </seg>
    <seg type="syll">qui </seg>
  </seg>
  <seg type="foot">
    <seg type="syll">pri</seg>
    <seg type="syll">mus </seg>
    <seg type="syll">ab </seg>
  </seg>
  <seg type="foot">
    <seg type="syll">or</seg>
    <seg type="syll">is </seg>
  </seg>
</l>
```

Source: [202]

The <seg> element may be used to identify any subcomponent of a line which has content; its type attribute may characterize such units in any way appropriate to the needs of the encoder. For the specific case of labeling each foot with its formal type ('dactyl', 'spondee', etc.), and each syllable with its metrical or prosodic status (syllables bearing primary or secondary stress, long syllables, short syllables), however, the specialized attributes **met** and **real** are defined, which provide a more systematic framework than the **type** attribute; see section *6.3. Rhyme and Metrical Analysis* below.

In classical verse, a hexameter like that above may also be formally divided into two cola or 'hemistiches'. This example provides a typical case, in that the boundary of the first colon falls in the middle of one of the feet (between the syllables 'no' and 'Tro'). If both kinds of segmentation are required, the **part** attribute might be used to mark the overlapping structure as follows.

```
<l>
  <seg type="hemistich">
    <seg type="foot">
      <seg type="syll">Ar</seg>
      <seg type="syll">ma </seg>
      <seg type="syll">vi</seg>
    </seg>
    <seg type="foot">
      <seg type="syll">rum</seg>
      <seg type="syll">que </seg>
      <seg type="syll">ca</seg>
    </seg>
    <seg type="foot" part="I">
      <seg type="syll">no </seg>
    </seg>
  </seg>
  <seg type="hemistich">
    <seg type="foot" part="F">
```

```
    <seg type="syll">Tro</seg>
   </seg>
   <seg type="foot">
    <seg type="syll">iae </seg>
    <seg type="syll">qui </seg>
   </seg>
  </seg>
</l>
```

Source: [202]

Instead of using the **part** attribute on the <seg> element, it might be simpler just to mark the point at which the caesura occurs. An additional element is provided for analyses of this kind, in which what is to be marked are points 'between the words', which have some significance within a verse line:

<caesura/> marks the point at which a metrical line may be divided.

In classical prosody, the caesura, which occurs within a foot, is distinguished from a diaeresis, which occurs on a foot boundary (not to be confused with the division of a diphthong into two syllables, or the diacritic symbol used to indicate such division, each of which is also termed diaeresis). This distinction is rarely made nowadays, the term *caesura* being used for any division irrespective of foot boundaries. No special-purpose <diaeresis> element is therefore provided.

As an example of the <caesura> element, we refer again to the example from Langland. An encoder might choose simply to record the location of the caesura within each line, rather than encoding each half-line as a segment in its own right, as follows:

```
<l>In a somer seson, <caesura/> whan softe was the sonne, </l>
<l>I shoop me into shroudes <caesura/> as I a sheep were, </l>
<l>In habite as an heremite <caesura/> unholy of werkes, </l>
<l>Went wide in this world <caesura/> wondres to here. </l>
```

Source: [118]

Logically, the opposite of caesura might be considered to be enjambement. When the verse module is included in a schema, an additional class called att.enjamb is defined as follows:

att.enjamb (enjambement) groups elements bearing the **enjamb** attribute.

@**enjamb** (enjambement) indicates that the end of a verse line is marked by enjambement.

The following lines demonstrate the use of the **enjamb** attribute to mark places where there is a discrepancy between the boundaries of the <l> elements and the syntactic structure of the verse (a discrepancy of some significance in some schools of verse):

```
<l enjamb="y">Un astrologue, un jour, se laissa choir</l>
<l>Au fond d'un puits.</l>
```

Source: [114]

6.3 Rhyme and Metrical Analysis

When the module for verse is in use, the following additional attributes are available to record information about rhyme and metrical form:

att.metrical defines a set of attributes which certain elements may use to represent metrical information.

@**met** (metrical structure, conventional) contains a user-specified encoding for the conventional metrical structure of the element.

@**real** (metrical structure, realized) contains a user-specified encoding for the actual realization of the conventional metrical structure applicable to the element.

@**rhyme** (rhyme scheme) specifies the rhyme scheme applicable to a group of verse lines.

These attributes may be attached to the <lg> element, or to the higher-level text-division elements , <div1>, etc. In general, the attributes should be specified at the highest level possible; they may not however be specifiable at the highest level if some of the subdivisions of a text are in prose and others in verse. All these attributes may also be attached to the <l> and <seg> elements, but the default notation for the **rhyme** attribute has no defined meaning when specified on <l> or <seg>. The value for these attributes may take any form desired by the encoder, but the nature of the notation used will determine how well the attribute values can be processed by automatic means.

The primary function of the metrical attributes is to encode the conventional metrical or rhyming structure within which the poet is working, rather than the actual prosodic realization of each line; the latter can be recorded using the **real** attribute, as further discussed below. A simple mechanism is also provided for recording the actual realization of a rhyme pattern; see *6.4. Rhyme*.

6.3.1 Sample Metrical Analyses

As a simple example of the use of these attributes, consider the following lines from Pope's 'Essay on Criticism':

```
<div
   type="book"
   n="1"
   met="-+|-+|-+|-+|-+/"
   rhyme="aa">
  <lg  n="1"  type="paragraph">
   <l>'Tis hard to say, if greater Want of Skill</l>
   <l>Appear in <hi>Writing</hi> or in <hi>Judging</hi> ill;</l>
   <l>But, of the two, less dang'rous is th'Offence,</l>
   <l>To tire our <hi>Patience</hi>, than mis-lead our <hi>Sense</hi>:</l>
  </lg>
</div>
```

Source: [159]

This text is written entirely in heroic couplets; each line is an iambic pentameter (which, using a common notation, can be described with the formula -+|-+|-+|-+|-+/, each - denoting a metrically unstressed syllable, each + a metrically stressed one, each | a foot boundary, and the / a line-end), and the couplets rhyme (which can be represented with the conventional formula **aa**).

Because both rhyme pattern and metrical form are consistent throughout the poem, they may be conveniently specified on the element; the values given for the attributes will be inherited by any metrical unit contained within the elements of this poem, and must be interpreted in the appropriate way.

Since the notation used in the **met**, **real**, and **rhyme** attributes is user-defined, no binding description can be given of its details or of how its interpretation must proceed. (A default notation is provided for the **rhyme** attribute, which however the encoder can replace with another; see section *6.4. Rhyme*.) It is expected, however, that software should be able to support these attributes in useful ways; the more intelligent the software is, and the more knowledge of metrics is built into it, the better it will be able to support these attributes. In the extract given above, for example, the **met** and **rhyme** attribute values specified on the element are inherited directly by the <lg> elements nested within it. Since the **met** value specifies the metrical form of a single verse line, the structure of the <lg> as a whole is understood to involve as many repetitions of the pattern as there

are lines in the verse paragraph. The same attribute value, when inherited in turn by the <l> element, must be understood *not* to repeat. With sufficiently sophisticated software, segments within the line might even be understood as inheriting precisely that portion of the formula which applies to the segment in question; this will, however, be easier to accomplish for some languages than for others.

The **rhyme** attribute in this example uses the default notation to specify a rhyme scheme applicable only to pairs of lines. As elsewhere, the default notation for the **rhyme** attribute has no meaning for metrical units at the line level or below. In verse forms where line-internal rhyme is structurally significant, e.g. in some skaldic poetry, the default notation is incapable of expressing the required information, since the rhyme pattern may need to be specified for units smaller than the line. In such cases, a user-specified rhyme notation must be substituted for the default notation, or else the rhyme pattern must be described using some alternative method (e.g. by using the <link> mechanism described below).

The precise semantics of the **met** attribute and the inferences which software is expected or able to draw from it, are implementation-dependent; so are the semantics and processing of the **rhyme** attribute, when user-specified notations are used.

A formal definition of the significance of each component of the pattern given as the value of the **met** attribute may be provided in the <metDecl> element within the <encodingDesc> element in the TEI header (see section *6.5. Metrical Notation Declaration*). The encoder is free to invent any notation appropriate to his or her analytic needs, provided that it is adequately documented in this element. The notation may define metrical components using invented or traditional names (such as 'iamb' or 'hexameter') or in terms of basic units such as codes for stressed or unstressed syllables, or a combination of the two.

The **real** (for 'realization') attribute may optionally be specified to indicate any deviation from the pattern defined by the **met** attribute which the encoder wishes to record. By default, the **real** attribute has the same value as the **met** attribute on the same element; it is only necessary to provide an explicit value when the realization differs in some way from the abstract metrical pattern. The tension between conventional metrical pattern and its realization may thus be recorded explicitly. For example, many readers of the above passage would stress the word 'But' at the beginning of the third line rather than the word 'of' following it, as the metrical pattern would normally require. This variation might be encoded as follows:

```
<l  real="+-|-+|-+|-+|-+">But, of the two, ...</l>
```

Where the **real** attribute is used to over-ride the default or conventional metrical pattern, it applies only to the element on which it is specified. The default pattern for any subsequent lines is unaffected.

As it happens, this particular kind of variation is very common in the English iambic pentameter — it even has a name: trochaic substitution — an encoder might therefore

choose to regard this not as an instance of a variant realization, but as an instance of a variant metrical form:

```
<l  met="+-|-+|-+|-+|-+">But, of the two, ...</l>
```

Alternatively, a different metrical notation might be defined, in which this kind of variation was permitted throughout the text.

In choosing whether to over-ride a metrical specification in this way or by using the **real** attribute, the encoder is required to determine whether the change is a systematic or conventional one (as in this example) or an occasional variation, perhaps for local effect. In the following example, from Goethe's 'Auf dem See', the variation is a matter of local realization:

```
<lg
   type="chevy-chase-stanza"
   met="-+-+-+-+/-+-+-+"
   rhyme="ababcdcd">
  <l  n="1"> Und frische Nahrung, neues Blut</l>
  <l  n="2"  real="+--+-+"> Saug' ich aus freier Welt;</l>
  <l  n="3"  real="+--+-+-+"> Wie ist Natur so hold und gut,</l>
  <l  n="4"  real="---+-+"> Die mich am Busen hält!</l>
  <l  n="5"> Die Welle wieget unsern Kahn</l>
  <l  n="6"> Im Rudertakt hinauf,</l>
  <l  n="7"> Und Berge, wolkig himmelan,</l>
  <l  n="8"> Begegnen unserm Lauf.</l>
</lg>
```

Source: [89]

On the other hand, the famous inserted alexandrine in Pope's 'Essay on Criticism', might be encoded as follows:

```
<l  n="356"> A needless alexandrine ends the song, </l>
<l  n="357"  met="-+|-+|-+|-+|-+|-+"  real="++|-+|-+|+-|++|-+"> That, like a wounded snake, drags its slow length
along.
</l>
```

Here the **met** attribute indicates that a different metrical convention (the alexandrine) is in force, while the **real** attribute indicates that there is a variation from that convention. As with many other aspects of metrical analysis, however, this is of necessity an entirely interpretive judgment.

6.3.2 Segment-Level versus Line-level Tagging

The examples given so far have encoded information about the realization of metrical conventions at the level of the whole verse-line. This has obvious advantages of simplicity, but the disadvantage that any deviation from metrical convention is not marked at its precise point of occurrence in the text. Greater precision may be achieved, but only at the cost of marking deviant metrical units explicitly. This may be done with the <seg> element, giving the variant realization as the value of the **real** attribute on that element. Using this method, the example given immediately above might be encoded as follows:

```
<l  n="356"> A need<seg  type="foot"  n="2"  real="--">less
    a</seg>lexandrine ends the song,</l>
<l  n="357"  met="-+|-+|-+|-+|-+|-+">
  <seg  n="1"  real="++"> That, like </seg> a wounded snake,
<seg  n="4"  real="+-"> drags its </seg>
  <seg  n="5"  real="++"> slow length </seg>
along.

</l>
```

The marking of the foot boundaries with the symbol | in the **met** attribute value of the <l> element allows the human reader, or a sufficiently intelligent software program, to isolate the correct portion of that attribute value as the default value for the same attribute on the <seg> elements for feet, namely -+. It is of course up to the encoder to decide whether or not to include the **n** attribute of <seg> here, and whether or not also to tag the feet in the line in which there is no deviation from the metrical convention. The ability of software to infer which foot is being marked, if not all are tagged, will depend heavily on the language of the text and the

knowledge of prosody built into the software; the fuller and more explicit the markup, the easier it will be for software to handle it. It may prove useful, however, to mark metrical deviations in the manner shown, even if the available software is not sufficiently intelligent to scan lines without aid from the markup. Human readers who are interested in prosody may well be able to exploit the markup in useful ways even with less sophisticated software.

There are circumstances where it may also be useful to use the **met** attribute of <seg>. If we wish to identify the exact location of the different types of foot in the first line of Virgil's *Aeneid*, the text could be encoded as follows (for simplicity's sake the caesura has been omitted):

```
<l>
  <seg  type="foot"  met="+--">Arma vi</seg>
  <seg  met="+--">rumque ca</seg>
  <seg  met="++">no Tro</seg>
  <seg  met="++">iae qui </seg>
  <seg  met="+--">primus ab</seg>
  <seg  met="++"> oris</seg>
</l>
```

An appropriate value of the **met** attribute might also be supplied on the enclosing element, to indicate that each foot may be made up of a dactyl or a spondee, so that the values given here for **met** at the level of the foot may be considered a series of local variations on this fundamental pattern; in cases like this, of course, the local variations may also be considered aspects of realization rather than of convention, in which case the **real** attribute may be used instead of **met**, if desired.

6.3.3 Metrical Analysis of Stanzaic Verse

The method described above may be used to encode quite complex verse forms, for instance various kinds of fixed-form stanzas. Let us take one of Dante's canzoni, in which each stanza except the last has the same combination of eleven-syllable and seven-syllable lines, and the same rhyme scheme:

```
<div
    type="canzone"
    met="E/E/S/E/S/E/E/S/E/S/E/S/S/E/S/E/E/S/S/E/E"
    rhyme="abbcdaccbdceeffghhhgg">
  <lg  n="1"  type="stanza">
    <l  n="1">Doglia mi reca nello core ardire</l>
  </lg>
</div>
```

Source: [2]

Here the **met** attribute specifies a metrical pattern for each of the twenty-one lines making up a stanza of the canzone. Each stanza inherits this definition from the parent element. The **rhyme** attribute specifies a rhyme scheme for each stanza, in the same way.

In the metrical notation used here, the letter **E** represents a line containing nine syllables which may or may not be metrically prominent, a tenth which is prominent and an optional non-prominent eleventh syllable. The letter 'S' is used to represent a line containing five syllables which may or may not be metrically prominent, a sixth which is prominent and an optional non-prominent seventh syllable. A suitable definition for this notation might be given by a <metDecl> element like the following:

```
<metDecl type="met" pattern="((E|S)/)+)">
  <metSym value="E" terminal="false">xxxxxxxxx+o</metSym>
  <metSym value="S" terminal="false">xxxxx+o</metSym>
  <metSym value="x">metrically prominent or non-prominent</metSym>
  <metSym value="+">metrically prominent</metSym>
  <metSym value="o">optional non prominent</metSym>
  <metSym value="/">line division</metSym>
</metDecl>
```

As noted above, the metrical pattern specified on the applies to each <lg> (stanza) element contained within the . In fact however, after seven stanzas of this type, there is a final stanza, known as a commiato or envoi, which follows a different metrical and rhyming scheme. The solution to this problem is simply to specify a new **met** attribute on the eighth stanza itself, which will override the default value inherited from parent , as follows:

```
<div met=".....">
  <lg>
    <l> ... </l>
  </lg>
  <lg type="commiato" met="E/S/S/E/S/E/E/S/S/E/E" rhyme="abbccdeeedd">
    <l n="1">Canzone, presso di qui è une donna</l>
  </lg>
</div>
```

Source: [2]

Note that, in the same way as for the **real** attribute, over-riding of this kind does not affect subsequent elements at the same hierarchic level. Any <lg> element following the commiato above would be assumed to use the same metrical and rhyming scheme as the one preceding the commiato. Moreover, although it is quite regular (in the sense that the last stanza of each canzone is a commiato), the over-riding must be specified for each case.

6.4 Rhyme

The **rhyme** attribute is used to specify the rhyme pattern of a verse form. It should not be confused with the <rhyme> element, which is used to mark the actual rhyming word or words:

<rhyme> marks the rhyming part of a metrical line.

Like the **met** attribute, the **rhyme** attribute can be used with a user-specified notation documented by the <metDecl> element in the TEI header. Unlike **met**, however, the **rhyme** attribute has a default notation; if this default notation is used, no <metDecl> element need be given.

The default notation for rhyme offers the ability to record patterns of rhyming lines, using the traditional notation in which distinct letters stand for rhyming lines. For a work in rhyming couplets, like the Pope example above, the **rhyme** attribute simply specifies **aa**, indicating that pairs of adjacent lines rhyme with each other. For a slightly more complex scheme, applicable to groups of four lines, in which lines 1 and 3 rhyme, as do lines 2 and 4, this attribute would have the value **abab**. The traditional Spenserian stanza has the pattern **ababbcbcc**, indicating that within each nine line stanza, lines 1 and 3 rhyme with each other, as do lines 2, 4, 5 and 7, and lines 6, 8 and 9.

Non-rhyming lines within such a group may be represented using a hyphen or an x, as in the following example:

```
<!-- example needed -->
```

The <rhyme> element may be used to mark the words (or parts of words) which rhyme according to a predefined pattern:

```
<lg  type="couplet"  rhyme="aa">
  <l>Outside in the distance a wildcat did <rhyme>growl</rhyme>
  </l>
  <l>Two riders were approaching and the wind began to <rhyme>howl</rhyme>
  </l>
</lg>
```

Source: [65]

The **label** attribute is used to specify which parts of a rhyme scheme a given set of rhyming words represent:

```
<lg  type="quatrain"  rhyme="abab">
  <l>I wander thro' each charter'd <rhyme  label="a">street</rhyme>,</l>
  <l>Near where the charter'd Thames does <rhyme  label="b">flow</rhyme>,</l>
  <l>And mark in every face I <rhyme  label="a">meet</rhyme>
  </l>
  <l>Marks of weakness, marks of <rhyme  label="b">woe</rhyme>.</l>
</lg>
<lg  rhyme="abab">
  <l>In every cry of every <rhyme  label="a">Man</rhyme>
  </l>
  <l>In every Infant's cry of <rhyme  label="b">fear</rhyme>,</l>
  <l>In every voice, in every <rhyme  label="a">ban</rhyme>,</l>
  <l>The mind-forg'd manacles I <rhyme  label="b">hear</rhyme>.</l>
</lg>
```

Source: [16]

Within a given scope, all <rhyme> elements with the same value for their **label** attribute are assumed to rhyme with each other: thus, in the above example, the two rhymes labelled `a` in the first stanza rhyme with each other, but not necessarily with those labelled `a` in the second stanza. The scope is defined by the nearest ancestor element for which the **rhyme** attribute has been supplied.

The <rhyme> element can appear anywhere within a verse line, and not necessarily around a single word. It can thus be used to mark quite complex internal rhyming schemes, as in the following example:

```
<lg  rhyme="ABCCBBA">
  <l>The sunlight on the <rhyme  label="A">garden</rhyme>
  </l>
  <l>
    <rhyme  label="A">Harden</rhyme>s and grows <rhyme  label="B">cold</rhyme>,</l>
  <l>We cannot cage the <rhyme  label="C">minute</rhyme>
  </l>
  <l>Wi<rhyme  label="C">thin it</rhyme>s nets of <rhyme  label="B">gold</rhyme>
  </l>
  <l>When all is <rhyme  label="B">told</rhyme>
  </l>
  <l>We cannot beg for <rhyme  label="A">pardon</rhyme>.</l>
</lg>
```

Source: [133]

This mechanism, although reasonably simple for simple cases, may not be appropriate for more complex applications. In general, rhyme may be considered as a special form of 'correspondence', and hence encoded using the mechanisms defined for that purpose in section *16.4. Correspondence and Alignment*. Similar considerations apply to other metrical features such as alliteration or assonance.

To use the correspondence mechanisms to represent the complex rhyming pattern of the above example, each <rhyme> element must be given a unique identifier, as follows:

```
<lg  rhyme="AB-BBA">
  <l>The sunlight on the <rhyme  xml:id="V-A1">garden</rhyme>
  </l>
  <l>
    <rhyme  xml:id="V-A2">Harden</rhyme>s and grows <rhyme  xml:id="V-B1">cold,</rhyme>
  </l>
  <l>We cannot cage the <rhyme  xml:id="V-C1">minute</rhyme>
  </l>
  <l>Wi<rhyme  xml:id="V-C2">thin it</rhyme>s nets of <rhyme  xml:id="V-B2">gold</rhyme>
  </l>
  <l>When all is <rhyme  xml:id="V-B3">told</rhyme>
  </l>
  <l>We cannot beg for <rhyme  xml:id="V-A3">pardon</rhyme>.</l>
</lg>
```

Now that each rhyming word, or part-word, has been tagged and allocated an arbitrary identifier, the general purpose <link> element may be used to indicate which of the <rhyme> elements share the same rhyme, as follows:

```
<linkGrp  type="rhyme">
  <link  targets="#V-A1 #V-A2 #V-A3"/>
  <link  targets="#V-B1 #V-B2 #V-B3"/>
  <link  targets="#V-C1 #V-C2"/>
</linkGrp>
```

For further discussion of the <link> and <linkGrp> element, see section *16.4. Correspondence and Alignment*.

The <rhyme> and <caesura> phrase level elements are made available by the model.lPart class when the module defined by this chapter is included in a schema.

6.5 Metrical Notation Declaration

When the module defined in this chapter is included in a schema, a specialised element is optionally available in the <encodingDesc> element of the TEI Header to document the metrical notation used in marking up a text.

<metDecl> (metrical notation declaration) documents the notation employed to represent a metrical pattern when this is specified as the value of a **met**, **real**, or **rhyme** attribute on any structural element of a metrical text (e.g. <lg>, <l>, or <seg>).

@**pattern** (regular expression pattern) specifies a regular expression defining any value that is legal for this notation.

<metSym> (metrical notation symbol) documents the intended significance of a particular character or character sequence within a metrical notation, either explicitly or in terms of other symbol elements in the same metDecl.

@**value** specifies the character or character sequence being documented.

@**terminal** specifies whether the symbol is defined in terms of other symbols (**terminal** is set to **false**) or in prose (**terminal** is set to **true**).

As with other components of the header, metrical notation may be specified either formally or informally. In a formal specification, every symbol used in the metrical notation must be documented by a corresponding <metSym> element; in an informal one, only a brief prose description of the way in which the notation is used need be given. In either case, the optional **pattern** attribute may be used to supply a regular expression which a processor can use to validate expressions in the intended notation. The following constraints apply:

- if **pattern** is supplied, any notation used which does not conform to it should be regarded as invalid
- if any <metSym> is defined, then any notation using undefined symbols should be regarded as invalid
- if both pattern and symbol are defined, then every symbol appearing explicitly within pattern must be defined
- symbols which are not matched by **pattern** may be defined within a <metDecl> element

As a simple example, consider the case of the notation in which metrical prominence, metrical feet, and line boundaries are all to be encoded. Legal specifications in this notation may be written for any sequence of metrically prominent or non-prominent features, optionally separated by foot or metrical line boundaries at arbitrary points. Assuming that the symbol *1* is used for metrical prominence, *0* for non-prominence, | for foot boundary and / for line boundary, then the following declaration achieves this object:

```
<metDecl pattern="((1|0)+\|?/?)*">
  <metSym value="1">metrical prominence</metSym>
  <metSym value="0">metrical non-prominence</metSym>
  <metSym value="|">foot boundary</metSym>
  <metSym value="/">metrical line boundary</metSym>
</metDecl>
```

The same notation might also be specified less formally, as follows:

```
<metDecl>
  <p>Metrically prominent syllables are marked '1' and other
     syllables '0'. Foot divisions are marked by a vertical bar,
     and line divisions with a solidus.</p>
  <p>This notation may be applied to any metrical unit, of any
     size (including, for example, individual feet as well as
     groups of lines).</p>
</metDecl>
```

Note that in this case, because the **pattern** attribute has not been supplied, no processor can validate **met** attribute values within the text which use this metrical notation.

For more complex cases, it will often be more convenient to define a notation incrementally. The **terminal** attribute should be used to indicate for a given symbol whether or not it may be re-defined in terms of other symbols used within the same notation. For example, here is a notation for encoding classical metres, in which symbols are provided for the most common types of foot. These symbols are themselves documented within the same notation, in terms of more primitive long and short syllables:

```
<metDecl pattern="[DTIS3A]+">
  <metSym n="dactyl" value="D" terminal="false">-oo</metSym>
```

```
  <metSym  n="trochee"  value="T"  terminal="false">-o</metSym>
  <metSym  n="iamb"  value="I"  terminal="false">o-</metSym>
  <metSym  n="spondee"  value="S"  terminal="false">--</metSym>
  <metSym  n="tribrach"  value="3"  terminal="false">ooo</metSym>
  <metSym  n="anapaest"  value="A"  terminal="false">oo-</metSym>
  <metSym  value="o">short syllable</metSym>
  <metSym  value="-">long syllable</metSym>
</metDecl>
```

Note here the use of the global **n** attribute to supply an additional name for the symbols being documented.

6.6 Encoding Procedures for Other Verse Features

A number of procedures that may be of particular concern to encoders of verse texts are dealt with elsewhere in these guidelines. Some aspects of layout and physical appearance, especially important in the case of free verse, are dealt with in chapter *11. Representation of Primary Sources*. Some initial recommendations for the encoding of phonetic or prosodic transcripts, which may be helpful in the analysis of sound structures in poetry, are to be found in chapter *8. Transcriptions of Speech*; it may also be found convenient to use standard entity names (those proposed for the International Phonetic Alphabet suggest themselves) to mark positions of suprasegmentals such as primary and secondary stress, or other aspects of accentual structure.

As already indicated, chapter *16. Linking, Segmentation, and Alignment* contains much which will be found useful for the aligning of multiple levels of commentary and structure within verse analysis. Encoders of verse (as of other types of literary text) will frequently wish to attach identifying labels to portions of text that are not part of a system of hierarchical divisions, may overlap with one another, and/or may be discontinuous; for instance passages associated with particular characters, themes, images, allusions, topoi, styles, or modes of narration. Much of the computerized analysis of verse seems likely to require dividing texts up into blocks in this way. The element discussed in *17.3. Spans and Interpretations* provides the means for doing this. Finally, the procedures for the tagging of feature structures, described in chapter *18. Feature Structures*, provide a powerful means of encoding a wide variety of aspects of verse literature, including not only the metrical structures discussed above, but also such stylistic and rhetorical features as metaphor.

For other features it must for the time being be left to encoders to devise their own terminology. Elements such as **<metaphor tenor="..." vehicle="...">** ... **</metaphor>** might well suggest themselves; but given the problems of definition involved, and the great richness of modern metaphor theory, it is clear that any such format, if predefined by these Guidelines, would have seemed objectionable to some and excessively restrictive to many. Leaving the choice of tagging terminology to individual encoders carries with it one vital corollary, however: the encoder must be utterly explicit, in the TEI header, about the methods of tagging used and the criteria and definitions on which they rest. Where no formal elements are currently proposed, such information may readily be given as simple prose description within the <encodingDesc> element defined in section *2.3. The Encoding Description*.

6.7 Module for Verse

The module described in this chapter makes available the following components:

Module verse: Verse structures

- Elements defined: caesura metDecl metSym rhyme
- Classes defined: att.enjamb att.metrical

The selection and combination of modules to form a TEI schema is described in *1.2. Defining a TEI Schema*.

Chapter 7

Performance Texts

This module is intended for use when encoding printed dramatic texts, screen plays or radio scripts, and written transcriptions of any form of performance.

Section *7.1. Front and Back Matter* discusses elements, such as cast lists, which can appear only in the front or back matter of printed dramatic texts. Section *7.2. The Body of a Performance Text* discusses the structural components of performance texts: these include major structural divisions such as acts and scenes (section *7.2.1. Major Structural Divisions*); individual speeches (section *7.2.2. Speeches and Speakers*); stage directions (section *7.2.3. Stage Directions*); and the elements making up individual speeches (section *7.2.4. Speech Contents*). Section *7.2.5. Embedded Structures* discusses ways of encoding units which cross the simple hierarchic structure so far defined, such as embedded songs or masques. Finally, section *7.3. Other Types of Performance Text* discusses a small number of additional elements characteristic of screen plays and radio or television scripts, as well as some elements for representing technical stage directions such as lighting or blocking.

The default structure for dramatic texts is similar to that defined by chapter *4. Default Text Structure*, as further discussed in section *7.2.1. Major Structural Divisions*.

Two element classes are used by this module. The model.frontPart.drama class supplies specialized elements which can appear only in the front or back matter of performance texts. The model.stageLike class supplies a set of elements for stage directions and similar items such as camera movements, which can occur between or within speeches.

7.1 Front and Back Matter

In dramatic texts, as in all TEI-conformant documents, the header element is followed by a <text> element, which contains optional front and back matter, and either a <body> or else a <group> of nested <text> elements. For more information on these, see chapter *4. Default Text Structure*.

The <front> and <back> elements are most likely to be of use when encoding preliminary materials in published performance texts. When the module defined by this chapter is included in a schema, the following additional elements not generally found in other forms of text become available as part of the front or back matter:

<performance> contains a section of front or back matter describing how a dramatic piece is to be performed in general or how it was performed on some specific occasion.

<prologue> contains the prologue to a drama, typically spoken by an actor out of character, possibly in association with a particular performance or venue.

<epilogue> contains the epilogue to a drama, typically spoken by an actor out of character, possibly in association with a particular performance or venue.

<set> (setting) contains a description of the setting, time, locale, appearance, etc., of the action of a play, typically found in the front matter of a printed performance text (not a stage direction).

<castList> (cast list) contains a single cast list or dramatis personae.

Elements for encoding each of these specific kinds of front matter are discussed in the remainder of this section, in the order given above. In addition, the front matter of dramatic texts may include the same elements as that of any other kind of text, notably title pages and various kinds of text division, as discussed in section *4.5. Front Matter*. The encoder may choose to ignore the specialized elements discussed in this section and instead use constructions of the type ** ** or **<div1 type="set">**.

Most other material in the front matter of a performance text will be marked with the default text structure elements described in chapter *4. Default Text Structure*. For example, the title page, dedication, other commendatory material, preface, etc., in a printed text should be encoded using or <div1> elements, containing headings, paragraphs, and other core tags.

7.1.1 The Set Element

A special form of note describing the setting of a dramatic text (that is, the time and place of its action) is sometimes found in the front matter.

<set> (setting) contains a description of the setting, time, locale, appearance, etc., of the action of a play, typically found in the front matter of a printed performance text (not a stage direction).

Descriptions of the setting may also appear as initial stage directions in the body of the play, but such descriptions should be marked as stage directions, not <set>. The <set> element should be used only where the description forms part of the front matter, as in the following examples:

```
<front>
  <castList>
    <castItem> ... </castItem>
  </castList>
  <set>
    <p>The action of the play is set in Chicago's
        Southside, sometime between World War II and the
        present.</p>
  </set>
</front>
```

Source: [97]

```
<front>
  <titlePage/>
  <div  type="copyright_page"/>
  <div  type="Contents"/>
  <div  type="Introduction"/>
  <div  type="note">
    <head>Note on the Translation</head>
    <p> ... </p>
  </div>
  <titlePage  type="half-title">
    <docTitle>
      <titlePart>Peer Gynt</titlePart>
    </docTitle>
  </titlePage>
  <div  type="Dramatis_Personae">
```

```
    <head>Characters</head>
    <castList/>
  </div>
  <set>
    <p>The action, which opens in the beginning of the nineteenth
        century, and ends around the 1860s, takes place partly in
        Gudbrandsdalen, and on the mountains around it, partly on the coast
        of Morocco, in the desert of Sahara, in a madhouse at Cairo, at sea,
        etc.</p>
  </set>
  <performance/>
</front>
```

Source: [101]

7.1.2 Prologues and Epilogues

Many plays in the Western tradition include in their front matter a prologue, spoken by an actor, generally not in character. Similar speeches often also occur at the end of the play, as epilogues. The elements <prologue> and <epilogue> are provided for the encoding of such features within the front or back matter, where appropriate.

<prologue> contains the prologue to a drama, typically spoken by an actor out of character, possibly in association with a particular performance or venue.

<epilogue> contains the epilogue to a drama, typically spoken by an actor out of character, possibly in association with a particular performance or venue.

A prologue may be encoded just like a distinct poem, as in the following example:

```
<front>
  <prologue>
    <head>Prologue, spoken by <name>Mr. Hart</name>
    </head>
    <l>Poets like Cudgel'd Bullys, never do</l>
    <l>At first, or second blow, submit to you;</l>
    <l>But will provoke you still, and ne're have done,</l>
    <l>Till you are weary first, with laying on:</l>
    <l>We patiently you see, give up to you,</l>
    <l>Our Poets, Virgins, nay our Matrons too.</l>
  </prologue>
  <castList>
    <head>The Persons</head>
    <castItem> ... </castItem>
  </castList>
  <set>
    <head>The SCENE</head>
    <p>London</p>
  </set>
</front>
```

Source: [212]

A prologue or epilogue may also be encoded as a speech, using the <sp> element described in section *3.12.2. Core Tags for Drama*. This is particularly appropriate where stage directions, etc., are involved, as in the following example:

```
<epilogue>
  <head>Written by <name>Colley Cibber, Esq</name>
     and spoken by <name>Mrs. Cibber</name>
  </head>
  <sp>
    <lg  type="stanza">
      <l>Since Fate has robb'd me of the hapless Youth,</l>
      <l>For whom my heart had hoarded up its truth;</l>
      <l>By all the Laws of Love and Honour, now,</l>
      <l>I'm free again to chuse, — and one of you</l>
    </lg>
    <lg  type="stanza">
      <l>Suppose I search the sober Gallery; — No,</l>
      <l>There's none but Prentices — & Cuckolds all a row:</l>
      <l>And these, I doubt, are those that make 'em so.</l>
    </lg>
    <stage>Pointing to the Boxes.</stage>
    <lg  type="stanza">
      <l>'Tis very well, enjoy the jest:</l>
    </lg>
  </sp>
</epilogue>
```

Source: [127]

In cases where the prologue or epilogue is clearly a significant part of the dramatic action, it may be preferable to include it in the body of a text, rather than in the front or back matter. In such cases, the encoder (and theatrical tradition) will determine whether or not to regard it as a new scene or division, or simply the final speech in the play. In the First Folio version of Shakespeare's *Tempest*, for example, Prospero's final speech is clearly marked off as a distinct textual unit by the headings and layout of the page, and might therefore be encoded as back matter:

```
<text>
  <body>
    <div1  type="scene">
      <sp>
        <l  part="Y">I'le deliver all,</l>
        <l>And promise you calme Seas, auspicious gales,</l>
        <l>Be free and fare thou well: please you, draw neere.</l>
        <stage>Exeunt omnes.</stage>
      </sp>
    </div1>
  </body>
  <back>
    <epilogue>
      <head>Epilogue, spoken by Prospero.</head>
      <sp>
        <l>Now my Charmes are all ore-throwne,</l>
        <l>And what strength I have's mine owne</l>
        <l>As you from crimes would pardon'd be,</l>
        <l>Let your Indulgence set me free.</l>
      </sp>
      <stage>Exit</stage>
    </epilogue>
    <set>
      <p>The Scene, an un-inhabited Island.</p>
```

```
    </set>
    <castList>
      <head>Names of the Actors.</head>
      <castItem>Alonso, K. of Naples</castItem>
      <castItem>Sebastian, his Brother.</castItem>
      <castItem>Prospero, the right Duke of Millaine.</castItem>
    </castList>
    <trailer>FINIS</trailer>
  </back>
</text>
```

Source: [177]

In many modern editions, the editors have chosen to regard Prospero's speech as a part of the preceding scene:

```
<sp>
  <speaker>Prospero</speaker>
  <l  part="Y">I'll deliver all,</l>
  <l>And promise you calm seas, auspicious gales,</l>
  <l>Be free and fare thou well. <stage  type="exit">Exit Ariel</stage>
     Please you, draw near. <stage  type="exit">Exeunt all but Prospero</stage>
    <note  place="margin">Epilogue</note>
  </l>
  <l>Now my charms are all o'erthrown,</l>
  <l>And what strength I have's mine own</l>
  <l>As you from crimes would pardoned be,</l>
  <l>Let your indulgence set me free.</l>
</sp>
<stage  type="mix">He awaits applause, then exit.</stage>
```

Source: [177]

7.1.3 Records of Performances

Performance texts are not only printed in books to be read, they are also performed. It is common practice therefore to include within the front matter of a printed dramatic text some brief account of particular performances, using the following element:

<performance> contains a section of front or back matter describing how a dramatic piece is to be performed in general or how it was performed on some specific occasion.

The <performance> element may be used to group any and all information relating to the actual performance of a play or screenplay, whether it specifies how the play should be performed in general or how it was performed in practice on some occasion.

Performance information may include complex structures such as cast lists, or paragraphs describing the date and location of a performance, details about the setting portrayed in the performance and so forth. (See the discussion of these specialized structures in section *7.1. Front and Back Matter* above.) If information for more than one performance is being recorded, then more than one <performance> element should be used, wherever possible.

Names of persons, places, and dates of particular significance within the performance record may be explicitly marked using the general purpose <name>, **<rs type="place">** and <date> elements described in section *3.5.4. Dates and Times*. No particular elements for such features as stagehouses, directors, etc., are proposed at this time.

For example:

```
<performance>
  <head>Death of a Salesman</head>
  <p>A New Play by Arthur Miller</p>
  <p>Staged by Elia Kazan</p>
  <castList>
    <head>Cast</head>
    <note  rend="small type flush left"  place="inline">(in order of appearance)</note>
    <castItem>
      <role>Willy Loman</role>
      <actor>Lee J. Cobb</actor>
    </castItem>
    <castItem>
      <role>Linda</role>
      <actor>Mildred Dunnock</actor>
    </castItem>
    <castItem>
      <role>Biff</role>
      <actor>Arthur Kennedy</actor>
    </castItem>
    <castItem>
      <role>Happy</role>
      <actor>Cameron Mitchell</actor>
    </castItem>
<!-- ... -->
  </castList>
  <p>The setting and lighting were designed by
  <name>Jo Mielziner</name>.</p>
  <p>The incidental music was composed by <name>Alex North</name>.</p>
  <p>The costumes were designed by <name>Julia Sze</name>.</p>
  <p>Presented by <name  rend="unmarked">Kermit Bloomgarden</name>
     and <name  rend="unmarked">Walter Fried</name> at the
  <rs  type="place">Morosco Theatre in New York</rs> on
  <date  when="1949-02-10">February 10, 1949</date>.</p>
</performance>
```

Source: [141]

Or:

```
<performance>
  <p>La Machine Infernale a été
     représentée pour la première fois au
  <rs  type="place-theatre">théâtre Louis-Jouvet</rs>
    <rs  type="place-theatre">(Comédie des
        Champs-élysées)</rs>
    <date>le 10 avril 1934</date>,
     avec les décors et les costumes de
  <name>Christian Bérard.</name> ... </p>
</performance>
```

Source: [40]

7.1.4 Cast Lists

A cast list is a specialized form of list, conventionally found at the start or end of a play, usually listing all the speaking and non-speaking roles in the play, often with additional description (‘Cataplasma, a maker of

Periwigges and Attires') or the name of an actor or actress ('Old Lady Squeamish. Mrs Rutter'). Cast lists may be encoded with the general purpose <list> element described in section *3.7. Lists*, but for more detailed work the following specialized elements are provided:

<castList> (cast list) contains a single cast list or dramatis personae.

<castGroup> (cast list grouping) groups one or more individual castItem elements within a cast list.

<castItem> (cast list item) contains a single entry within a cast list, describing either a single role or a list of non-speaking roles.

@**type** characterizes the cast item.

A <castItem> element may contain any mixture of elements taken from the model.castItemPart class, members of which (when this module is included) are:

<role> the name of a dramatic role, as given in a cast list.

<roleDesc> (role description) describes a character's role in a drama.

<actor> Name of an actor appearing within a cast list.

Cast lists often have an internal structure of their own; it is quite usual to find, for example, nobility and commoners, or male and female roles, presented in different groups or sublists. Roles are also often grouped together by their function, for example:

- Sons of Cato:
  - Portius
  - Marcus

A cast list relating to a specific performance may be accompanied by notes about the time or place of that performance, indicating (for example) the name of the theatre where the play was first presented, the name of the producer or director, and so forth. When the cast list relates to a specific performance, it should be embedded within a <performance> element (see section *7.1.3. Records of Performances*), as in the following example:

```
<performance>
  <p>The first performance in Great Britain of <title>Waiting for
        Godot</title> was given at the Arts Theatre, London, on
  <date  when="1955-08-03">3rd August 1955</date>. It was directed by
  <name>Peter Hall</name>, and the décor was by <name>Peter
        Snow</name>. The cast was as follows:</p>
  <castList>
    <castItem>Estragon: Peter Woodthorpe</castItem>
    <castItem>Vladimir: Paul Daneman</castItem>
    <castItem> ... </castItem>
  </castList>
</performance>
```

Source: [10]

In this example, the <castItem> elements have no substructure. If desired, however, their components may be more finely distinguished using the elements <role>, <roleDesc>, and <actor>. For example, the second cast item above might be encoded as follows:

```
<castItem>
  <role  xml:id="vlad">Vladimir</role>:
<actor>Paul Daneman</actor>
</castItem>
```

The global **xml:id** attribute may be used to specify a unique identifier for the <role> element, where it is desired to link speeches within the text explicitly to the role, using the **who** attribute, as further discussed in section *7.2.2. Speeches and Speakers* below.

The occasionally lengthy descriptions of a role sometimes found in written play scripts may be marked using the <roleDesc> element, as in the following example:

```
<castItem>
  <role>Tom Thumb the Great</role>
  <roleDesc>a little hero with a great soul, something violent in his
     temper, which is a little abated by his love for Huncamunca</roleDesc>
  <actor>Young Verhuyk</actor>
</castItem>
```

Source: [72]

For non-speaking or un-named roles, a <castItem> may contain a <roleDesc> without an accompanying <role>, for example

```
<castItem>
  <roleDesc>Costermonger</roleDesc>
</castItem>
```

When a list of such minor roles is given together, the **type** attribute of the <castItem> should indicate that it contains more than one role, by taking a value such as **list**. The encoder may or may not elect to encode each separate constituent within such a composite <castItem>. Thus, either of the following is acceptable:

```
<castItem  type="list">Constables, Drawer, Turnkey, etc.</castItem>
<castItem  type="list">
  <roleDesc>Constables,</roleDesc>
  <roleDesc>Drawer,</roleDesc>
  <roleDesc>Turnkey,</roleDesc>
etc.

</castItem>
```

A group of cast items forming a distinct subdivision of a cast list may be marked as such by using the special purpose <castGroup> element. The **rend** attribute may be used to indicate whether this grouping is indicated in the text by layout alone (i.e. the use of whitespace), by long braces or by some other means. A <castGroup> may contain an optional heading (represented as usual by a <head> element) followed by a series of <castItem> elements:

```
<castGroup  rend="braced">
  <head>friends of Mathias</head>
  <castItem>
    <role>Walter</role>
    <actor>Mr Frank Hall</actor>
  </castItem>
  <castItem>
    <role>Hans</role>
    <actor>Mr F.W. Irish</actor>
  </castItem>
</castGroup>
```

Source: [125]

Alternatively, the encoder may prefer to regard the phrase ‘friends of Mathias’ as a role description, and encode the above example as follows:

```
<castGroup  rend="braced">
  <roleDesc>friends of Mathias</roleDesc>
  <castItem>
    <role>Walter</role>
    <actor>Mr Frank Hall</actor>
  </castItem>
  <castItem>
    <role>Hans</role>
    <actor>Mr F.W. Irish</actor>
  </castItem>
</castGroup>
```

This version has the advantage that all role descriptions are treated alike, rather than in some cases being treated as headings. On the other hand there are also cases, such as the following, where the role description does function more like a heading:

```
<castList>
  <castGroup>
    <head  rend="braced">Mendicants</head>
    <castItem>
      <role>Aafaa</role>
      <actor>Femi Johnson</actor>
    </castItem>
    <castItem>
      <role>Blindman</role>
      <actor>Femi Osofisan</actor>
    </castItem>
    <castItem>
      <role>Goyi</role>
      <actor>Wale Ogunyemi</actor>
    </castItem>
    <castItem>
      <role>Cripple</role>
      <actor>Tunji Oyelana</actor>
    </castItem>
  </castGroup>
  <castItem>
    <role>Si Bero</role>
    <roleDesc>Sister to Dr Bero</roleDesc>
    <actor>Deolo Adedoyin</actor>
  </castItem>
  <castGroup>
    <head  rend="braced">Two old women</head>
    <castItem>
      <role>Iya Agba</role>
      <actor>Nguba Agolia</actor>
    </castItem>
    <castItem>
      <role>Iya Mate</role>
      <actor>Bopo George</actor>
    </castItem>
```

```
  </castGroup>
  <castItem>
    <role>Dr Bero</role>
    <roleDesc>Specialist</roleDesc>
    <actor>Nat Okoro</actor>
  </castItem>
  <castItem>
    <role>Priest</role>
    <actor>Gbenga Sonuga</actor>
  </castItem>
  <castItem>
    <role>The old man</role>
    <roleDesc>Bero's father</roleDesc>
    <actor>Dapo Adelugba</actor>
  </castItem>
</castList>
```

Source: [185]

7.2 The Body of a Performance Text

The body of a performance text may be divided into structural units, variously called acts, scenes, stasima, entr'actes, etc. All such formal divisions should be encoded using an appropriate text-division element (, <div1>, <div2>, etc.), as further discussed in section *7.2.1. Major Structural Divisions*. Whether divided up into such units or not, all performance texts consist of sequences of speeches (see *7.2.2. Speeches and Speakers*) and stage directions (see *7.2.3. Stage Directions*). Speeches will generally consist of a sequence of chunk-level items: paragraphs, verse lines, stanzas, or (in case of uncertainty as to whether something is verse or prose) <seg> elements (see *7.2.4. Speech Contents*).

The boundaries of formal units such as verse lines or paragraphs do not always coincide with speech boundaries. Units such as songs may be discontinuous or shared among several speakers. As described below in section *7.2.5. Embedded Structures*, such fragmentation may be encoded in a relatively simple fashion using the linkage and aggregation mechanisms defined in chapter *16. Linking, Segmentation, and Alignment*.

7.2.1 Major Structural Divisions

Large divisions in drama such as acts, scenes, stasima, or entr'actes are indicated by numbered or unnumbered elements, as described in section *4.1. Divisions of the Body*. The **type** and **n** attributes may be used to define the type of division being marked, and to provide a name or number for it, as in the following example:

```
<body>
  <div1 type="scene" n="1">
    <head>Night—Faust's Study (i)</head>
  </div1>
  <div1 type="scene" n="2">
    <head>Outside the City Gate</head>
  </div1>
</body>
```

Source: [88]

Where the largest divisions of a performance text are themselves subdivided, most obviously in the case of plays traditionally divided into acts and scenes, further nested text-division elements may be used, as in this example:

```
<body>
  <div1  type="act"  n="1">
    <head>Act One</head>
    <div2  type="scene"  n="1">
      <stage>Pa Ubu, Ma Ubu</stage>
      <sp>
        <speaker>Pa Ubu</speaker>
        <p>Pschitt!</p>
      </sp>
    </div2>
    <div2  type="scene"  n="2">
      <stage>A room in Pa Ubu's house, where a magnificent
           collation is set out</stage>
    </div2>
  </div1>
  <div1  type="act"  n="2">
    <head>Act Two</head>
    <div2  type="scene"  n="1">
      <head>Scene One</head>
    </div2>
    <div2  type="scene"  n="2">
      <head>Scene Two</head>
    </div2>
  </div1>
</body>
```

Source: [105]

In the example above, the <div2> element has been used to represent the 'French scene' convention, (where the entrance of each new set of characters is marked as a distinct unit in the text) and the <div1> element to represent the acts into which the play is divided. The elements chosen are determined only by the hierarchic position of these units in the text as a whole. If the text had no acts, but only scenes, then the scenes might be represented by <div1> elements. Equally, if a play is divided only into 'acts', with no smaller subdivisions, then the <div1> element might be used to represent acts. The **type** should be used, as above, to make explicit the name associated with a particular category of subdivision.

As an alternative to the use of numbered divisions, the encoder may represent all subdivisions with the same element, the unnumbered . The second act in the above example would then be represented as follows:

```
<div  type="act"  n="2">
  <head>Act Two</head>
  <div  type="scene"  n="1">
    <head>Scene One</head>
  </div>
  <div  type="scene"  n="2">
    <head>Scene Two</head>
  </div>
</div>
```

For further discussion of the use of numbered and unnumbered divisions, see section *4.1. Divisions of the Body*.

7.2.2 Speeches and Speakers

The following elements are used to identify speeches and speakers in a performance text:

<sp> (speech) An individual speech in a performance text, or a passage presented as such in a prose or verse text.

<speaker> A specialized form of heading or label, giving the name of one or more speakers in a dramatic text or fragment.

As noted above, the structure of many performance texts may be analysed as multiply hierarchic: a scene of a verse play, for example, may be divided into speeches and, at the same time, into verse lines. The end of a line may or may not coincide with the end of a speech, and vice versa. Other structures, such as songs, may be discontinuous or split up over several speeches. For some purposes it will be appropriate to regard the verse-structure as the fundamental organizing principle of the text, and for others the speech structure; in some cases, the choice between the two may be arbitrary. The discussion in the remainder of this chapter assumes that it is the speech-based hierarchy which most prominently determines the structure of performance texts, but the same mechanisms could be employed to encode a view of a performance text in which individual speeches were entirely subordinate to the formal units of prose and verse. For more detailed discussion and examples of various treatments of this fundamental issue, refer to chapter *20. Non-hierarchical Structures*.

The **who** attribute and the <speaker> element are both used to indicate the speaker or speakers of a speech, but in rather different ways. The <speaker> element is used to encode the word or phrase actually used within the source text to indicate the speaker: it may contain any string or prefix, and may be thought of as a highly specialized form of stage direction. The value of the **who** attribute however is a unique code, probably made up by the transcriber, which will unambiguously identify the character to whom the speech is assigned. To enforce this uniqueness, the base tag set for drama defines the value of this attribute as **IDREFS**. This means that the codes included in it must correspond with codes which are specified elsewhere in the document as identifiers for particular elements, typically the <role> element in the cast list where the character is named or described, as discussed in *7.1. Front and Back Matter* above.

```
<castList>
  <castItem>
    <role xml:id="menae">Menaechmus</role>
  </castItem>
  <castItem>
    <role xml:id="penic">Peniculus</role>
  </castItem>
</castList>
<sp who="#menae">
  <speaker>Menaechmus</speaker>
  <l>Responde, adulescens, quaeso, quid nomen tibist?</l>
</sp>
<sp who="#penic">
  <speaker>Peniculus</speaker>
  <l>Etiam derides, quasi nomen non noveris?</l>
</sp>
<sp who="#menae">
  <speaker>Menaechmus</speaker>
  <l>Non edepol ego te, quot sciam, umquam ante hunc diem</l>
  <l>Vidi neque novi; ...</l>
</sp>
```

Source: [157]

If present, a <speaker> element may only appear as the first part of an <sp> element. The distinction between the <speaker> element and the **who** attribute makes it possible to encode uniformly characters whose names are not indicated in a uniform fashion throughout the play, or characters who appear in disguise, as in the following examples:

```
<castList>
  <castItem>
    <role xml:id="hh">Henry Higgins</role>
  </castItem>
</castList>
<sp who="#hh">
  <speaker>The Notetaker</speaker>
  <p> ... </p>
</sp>
```

Source: [180]

If the speaker attributions are completely regular (and may thus be reconstructed mechanically from the values given for the **who** attribute), or are of no interest for the encoder of the text (as might be the case with editorially supplied attributions in older texts), then the <speaker> element need not be used; the former example above then might look like this:

```
<castList>
  <castItem>
    <role xml:id="menaechmus">Menaechmus</role>
  </castItem>
  <castItem>
    <role xml:id="peniculus">Peniculus</role>
  </castItem>
</castList>
<sp who="#menaechmus">
  <l>Responde, adulescens, quaeso, quid nomen tibist?</l>
</sp>
<sp who="#peniculus">
  <l>Etiam derides, quasi nomen non noveris?</l>
</sp>
<sp who="#menaechmus">
  <l>Non edepol ego te, quot sciam, umquam ante hunc diem</l>
  <l>Vidi neque novi; ...</l>
</sp>
```

Source: [157]

More than one identifier may be listed as value for the **who** attribute if the speech is spoken by more than one person, as in the following example:

```
<castList>
  <castItem>
    <role xml:id="nan">Nano</role>
  </castItem>
  <castItem>
    <role xml:id="cas">Castrone</role>
  </castItem>
</castList>
<stage>Nano and Castrone sing</stage>
<sp who="#nan #cas">
  <l>Fools, they are the only nation</l>
  <l>Worth men's envy or admiration</l>
</sp>
```

Source: [107]

The <sp> and <speaker> elements are both declared within the core module (see section *3.12. Passages of Verse or Drama*).

7.2.3 Stage Directions

Both between and within the speeches of a written performance text, it is normal practice to include a wide variety of descriptive directions to indicate non-verbal action. The following elements are provided to represent these:

<stage> (stage direction) contains any kind of stage direction within a dramatic text or fragment.
- @**type** indicates the kind of stage direction.

<move/> (movement) marks the actual entrance or exit of one or more characters on stage.
- @**type** characterizes the movement, for example as an entrance or exit.
- @**where** specifies the direction of a stage movement.
- @**perf** (performance) identifies the performance or performances in which this movement occurred as specified.

A satisfactory typology of stage directions is difficult to define. Certain basic types such as 'entrance', 'exit', 'setting', 'delivery', are easily identified. But the list is not a closed one, and it is not uncommon to mix types within a single direction. No closed set of values for the **type** attribute is therefore proposed at the present time, though some suggested values are indicated in the list below, which also indicates the range of possibilities.

```
<stage type="setting">The throne descends.</stage>
<stage type="setting">Music</stage>
<stage type="entrance">Enter Husband as being thrown off his horse.</stage>
<stage type="exit">Exit pursued by a bear.</stage>
<stage type="business">He quickly takes the stone out.</stage>
<stage type="delivery">To Lussurioso.</stage>
<stage type="delivery">Aside.</stage>
<stage type="delivery">Not knowing what to say.</stage>
<stage type="costume">Disguised as Ansaldo.</stage>
<stage type="location">At a window.</stage>
<stage type="novelistic">Having had enough, and embarrassed
for the family.</stage>
```

The meaning of the values used for the **type** attribute on <stage> elements may be defined within the <tagUsage> element of the TEI header (described in section *2.3.4. The Tagging Declaration*). For example:

```
<tagUsage gi="stage">This element is used for all stage directions,
editorial or authorial. The type attribute on this element takes
one or more of the following values:
<list type="gloss">
   <label>setting</label>
   <item>describes the set</item>
   <label>blocking</label>
   <item>describes movement across stage, position, etc.</item>
   <label>business</label>
   <item>describes movement other than blocking</item>
   <label>delivery</label>
   <item>describes how the line is said</item>
   <label>motivation</label>
   <item>describes character's emotional state or through line</item>
```

```
  </list>
</tagUsage>
```

This approach is purely documentary; in a real project it would generally be more effective to define the range of permitted values explicitly within the project's schema specification, using the techniques described in chapter *23.2. Personalization and Customization.* For example, a specification like the following might be used to produce a schema in which the **type** attribute of the <stage> element is permitted to take only the values listed above:

```
<schemaSpec ident="myDrama">
  <moduleRef key="core"/>
  <moduleRef key="tei"/>
  <moduleRef key="structure"/>
  <moduleRef key="header"/>
  <moduleRef key="drama"/>
  <elementSpec ident="stage" mode="change">
    <attList>
      <attDef ident="type" mode="replace">
        <valList type="closed">
          <valItem ident="setting">
            <desc>describes the set</desc>
          </valItem>
          <valItem ident="blocking">
            <desc>describes movement across stage, position, etc.</desc>
          </valItem>
          <valItem ident="business">
            <desc>describes movement other than blocking</desc>
          </valItem>
          <valItem ident="delivery">
            <desc>describes how the line is said</desc>
          </valItem>
          <valItem ident="motivation">
            <desc>describes character's emotional state or through line</desc>
          </valItem>
        </valList>
      </attDef>
    </attList>
  </elementSpec>
</schemaSpec>
```

The <stage> element may appear both between and within <sp> elements. It may contain a mixture of phrase level elements, possibly combined into paragraphs, as in the following example:

```
<div1 n="1" type="act">
  <stage type="setting">
    <p>Scene. — A room furnished comfortably and
        tastefully but not extravagantly ...
        The floor is carpeted and a fire burns in the stove.
        It is winter.</p>
    <p>A bell rings in the hall; shortly afterwards the
        door is heard to open. Enter NORA humming a tune ...</p>
  </stage>
  <sp>
```

```
   <speaker>Nora</speaker>
   <p>Hide the Christmas Tree carefully, Helen. Be sure the
       children do not see it till this evening, when it is
       dressed. <stage type="delivery">To the PORTER taking
          out her purse</stage> How much?</p>
  </sp>
</div1>
```

Source: [102]

The <stage> element may also be used in non-theatrical texts, to mark sound effects or musical effects, etc., as further discussed in section *7.3. Other Types of Performance Text*.

The <move> element is intended to help overcome the fact that the stage directions of a printed text may often not provide full information about either the intended or the actual movement of actors on stage. It may be used to keep track of entrances and exits in detail, so as to know which characters are on stage at which time. Its attributes permit a relatively formal specification for movements of characters, using user-defined codes to identify the characters involved (the **who** attribute), the direction of the movement (**type** attribute), and optionally which part of the stage is involved (**where** attribute). For stage-historical purposes, a **perf** attribute is also provided; this allows the recording of different <move> elements as taken in different performances of the same text.

The <move> element should be located at the position in the text where the move is presumed to take place. This will often coincide with a stage direction, as in the following simple example:

```
<castList>
  <castItem>
    <role xml:id="bella">Bellafront</role>
  </castItem>
</castList>
<stage type="entrance">
  <move who="#bella" type="enter"/>
Enter Bellafront mad.
</stage>
```

Source: [56]

The <move> element can however appear independently of a stage direction, as in the following example:

```
<castList>
  <castItem>
    <role xml:id="lm">Lady Macbeth</role>
  </castItem>
  <castItem>
    <role xml:id="g1">First Gentleman</role>
  </castItem>
<!-- ... -->
</castList>
<sp who="#g1">
  <speaker>Gent.</speaker>
  <p>Neither to you, nor any one; having no witness
     to confirm my speech. <move who="#lm" type="enter" where="C"/>
     Lo you! here she comes. This is her very guise; and,
     upon my life, fast asleep.</p>
</sp>
```

Source: [173]

7.2.4 Speech Contents

The actual speeches of a dramatic text may be composed of running text, which must be formally organized into paragraphs, in the case of prose (see section *3.1. Paragraphs*), verse lines or line groups in that of verse (see section *3.12. Passages of Verse or Drama*), or <seg> elements, in case of doubt as to whether the material should be treated as verse or prose. The following elements, all of which are defined in the core, are particularly useful when marking units of prose or verse within speeches:

<p> (paragraph) marks paragraphs in prose.

<lb/> (line break) marks the start of a new (typographic) line in some edition or version of a text.

@**ed** (edition) indicates the edition or version in which the line break is located at this point

<l> (verse line) contains a single, possibly incomplete, line of verse.

@**part** specifies whether or not the line is metrically complete.

<lg> (line group) contains a group of verse lines functioning as a formal unit, e.g. a stanza, refrain, verse paragraph, etc.

As a member of the classes att.typed and att.divLike, the <lg> element also bears the following attributes:

att.typed provides attributes which can be used to classify or subclassify elements in any way.

@**type** characterizes the element in some sense, using any convenient classification scheme or typology.

@**subtype** provides a sub-categorization of the element, if needed

att.divLike provides attributes common to all elements which behave in the same way as divisions.

@**org** (organization) specifies how the content of the division is organized.

@**sample** indicates whether this division is a sample of the original source and if so, from which part.

@**part** specifies whether or not the division is fragmented by some other structural element, for example a speech which is divided between two or more verse stanzas.

When the verse module is included in a schema, the elements <l> and <lg> also gain additional attributes through their membership of the class att.metrical:

att.metrical defines a set of attributes which certain elements may use to represent metrical information.

@**met** (metrical structure, conventional) contains a user-specified encoding for the conventional metrical structure of the element.

@**rhyme** (rhyme scheme) specifies the rhyme scheme applicable to a group of verse lines.

In many texts, prose and verse may be inextricably mingled; particularly in earlier printed texts, prose may be printed as verse or verse as prose, or it may be impossible to distinguish the two. In cases of doubt, an encoder may prefer to tag the dubious material consistently as verse, to tag it all as prose, to follow the typography of the source text, or to use the neutral <ab> element to contain the speech itself. When this question arises, the <tagUsage> element in the <encodingDesc> element of the header may be used to record explicitly what policy has been adopted.

Even where they can reliably be distinguished, a single speech may frequently contain a mixture of prose (marked as <p>) and verse (marked as <l> or — if stanzaic — <lg>).

The **part** attribute of the <l> and <lg> elements provides one simple way of indicating where the boundaries of a speech and of a verse line or line group do not coincide. The encoder may simply indicate that a line or line group is metrically incomplete by specifying the value **Y** or **N**, as in the following example:

```
<sp>
  <speaker>Face</speaker>
  <l  part="N">You most
     notorious whelp, you insolent slave</l>
  <l  part="Y">Dare you do this?</l>
</sp>
<sp>
  <speaker>Subtle</speaker>
  <l  part="Y">Yes faith, yes faith.</l>
</sp>
<sp>
  <speaker>Face</speaker>
  <l  part="Y">Why!  Who</l>
  <l  part="Y">Am I, my mongrel? Who am I?</l>
</sp>
<sp>
  <speaker>Subtle</speaker>
  <l  part="Y">I'll tell you,</l>
<!-- ... -->
</sp>
```

Source: [108]

Alternatively, where the fragments of the line or line group are consecutive in the text (though possibly interrupted by stage directions), the values **I** (initial), **M** (medial), and **F** (final) may be used to indicate how metrical lines are constituted:

```
<sp>
  <speaker>Face</speaker>
  <l>You most
     notorious whelp, you insolent slave</l>
  <l  part="I">Dare you do this?</l>
</sp>
<sp>
  <speaker>Subtle</speaker>
  <l  part="M">Yes faith, yes faith.</l>
</sp>
<sp>
  <speaker>Face</speaker>
  <l  part="F">Why!  Who</l>
  <l  part="I">Am I, my mongrel? Who am I?</l>
</sp>
<sp>
  <speaker>Subtle</speaker>
  <l  part="F">I'll tell you,</l>
<!-- ... -->
</sp>
```

In dramatic texts, the <lg> or line group element is most often of use for the encoding of songs and other stanzaic material, as further discussed in the next section. Line groups may be fragmented across speakers in the same way as individual lines, and the same set of attributes is available to record this fact. In the following example, an <lg> element is used to represent one verse of a song, which is divided between several voices:

```
<stage type="head">Song — Sir Joseph</stage>
<sp>
  <lg type="song" part="I">
    <l>I am the monarch of the sea,</l>
    <l>The ruler of the Queen's Navee.</l>
    <l>Whose praise Great Britain loudly chants.</l>
  </lg>
</sp>
<sp>
  <speaker>Cousin Hebe</speaker>
  <lg type="song" part="M">
    <l>And we are his sisters and his cousins and his aunts!</l>
  </lg>
</sp>
<sp>
  <speaker>Rel.</speaker>
  <lg type="song" part="F">
    <l>And we are his sisters and his cousins and his aunts!</l>
  </lg>
</sp>
```

Source: [85]

These elements are all defined in the core, and are thus available to every TEI document without formality. A more detailed discussion of the encoding of verse is provided in chapter *6. Verse*.

7.2.5 Embedded Structures

Although primarily composed of speeches, performance texts often contain other structural units such as songs or strophes which are shared among different speakers. More generally, complex nested structures of plays within plays, interpolated masques, or interludes are far from uncommon. In more modern material, comparably complex structural devices such as flashback or nested playback are equally frequent. In all kinds of performance material, it may be necessary to indicate several actions which are happening simultaneously.

A number of different devices are available within the TEI scheme to support these complexities in the general case. Texts may be composite or self-nesting (see section *4.3.1. Grouped Texts*) and multiple hierarchies may be defined (see chapter *20. Non-hierarchical Structures*). The TEI encoding scheme provides a variety of linking mechanisms, which may be used to indicate temporal alignment and aggregation of fragmented structures. In this section we provide a few specific examples of the application of these techniques to performance texts:

- the use of the <floatingText> element
- the use of the **part** attribute on fragmentary <lg> elements
- the use of the **next** and **prev** attributes on fragments of embedded structures to join them into a larger whole
- the use of the <join> element to define a 'virtual element' composed of the fragments indicated

When the whole of a song appears within a single speech, it may require no special treatment if it is considered to form a part of the speech:

```
<sp>
  <speaker>Kelly</speaker>
  <stage>(calmly).</stage>
  <p>Aha, so you've bad minds along with th' love of gain.
```

```
    You thry to pin on others th' dirty decorations that
    may be hangin' on your own coats.</p>
  <stage>(He points, one after the other at Conroy, Bull,
    and Flagonson. Lilting)</stage>
  <lg  type="song">
    <l>Who were you with last night?</l>
    <l>Who were you with last night?</l>
    <l>Will you tell your missus when you go home</l>
    <l>Who you were with last night?</l>
  </lg>
</sp>
<sp>
  <speaker>Flagonson</speaker>
  <stage>(in anguished indignation).</stage>
  <p>This is more than a hurt to us: this hits at the
    decency of the whole nation!</p>
</sp>
```

Source: [149]

If however, the song is to be regarded as forming a distinct item, perhaps with its own front and back matter, it may be better to regard it as a floating text:

```
<sp>
  <speaker>Kelly</speaker>
  <stage>(calmly).</stage>
  <p>Aha, so you've bad minds along with ...</p>
</sp>
<stage>(He points, one after the other at Conroy, Bull,
and Flagonson. Lilting):</stage>
<floatingText>
  <front>
    <titlePart>Kelly's Song</titlePart>
  </front>
  <body>
    <l>Who were you with last night?</l>
    <l>Who were you with last night?</l>
    <l>Will you tell your missus when you go home</l>
    <l>Who you were with last night?</l>
  </body>
</floatingText>
```

Source: [149]

When an embedded structure extends across more than one <sp> element, each of its constituent parts must be regarded as a distinct fragment; the problem then facing the encoder is to reconstitute the interrupted whole in some way.

As already noted above, the **part** attribute may be used to indicate that an <l> element contains a partial, not a complete, verse line. The same attribute may be used on the <lg> element, to indicate that the line group is partial rather than complete, thus:

```
<sp>
  <speaker>Kelly</speaker>
  <stage>(wheeling quietly in his semi-dance,
    as he goes out):</stage>
```

```
  <lg  type="stanza"  part="I">
    <l>Goodbye to holy souls left here,</l>
    <l>Goodbye to man an' fairy;</l>
  </lg>
</sp>
<sp>
  <speaker>Widda Machree</speaker>
  <stage>(wheeling quietly in her semi-dance,
     as she goes out):</stage>
  <lg  type="stanza"  part="F">
    <l>Goodbye to all of Leicester Square,</l>
    <l>An' the long way to Tipperary.</l>
  </lg>
</sp>
```

Source: [149]

When the fragments of a song are separated by other intervening dialogue, or even when not, they may be linked together with the **next** and **prev** attributes defined in section *16.7. Aggregation*. For example, the line groups making up Ophelia's song might be encoded as follows:

```
<div1  n="4"  type="act">
  <div2  n="5"  type="scene">
    <stage>Elsinore. A room in the Castle.</stage>
    <stage  type="setting">Enter Ophelia, distracted.</stage>
    <sp>
      <speaker>Ophelia</speaker>
      <p>Where is the beauteous Majesty of Denmark?</p>
    </sp>
    <sp>
      <speaker>Queen</speaker>
      <p>How now, Ophelia?</p>
    </sp>
    <sp>
      <speaker>Ophelia</speaker>
      <stage>Singing</stage>
      <lg
         next="#Tl2"
         xml:id="Tl1"
         type="song"
         part="Y">
        <l>How should I your true-love know</l>
        <l>From another one?</l>
        <l>By his cockle hat and staff</l>
        <l>And his sandal shoon.</l>
      </lg>
    </sp>
    <sp>
      <speaker>Queen</speaker>
      <p>Alas, sweet lady, what imports this song?</p>
    </sp>
    <sp>
      <speaker>Ophelia</speaker>
      <p>Say you? Nay, pray you mark.</p>
      <stage>Sings</stage>
      <lg
```

```
        prev="#Tl1"
        xml:id="Tl2"
        type="song"
        part="Y">
       <l>He is dead and gone, lady,</l>
       <l>He is dead and gone;</l>
       <l>At his head a grass-green turf,</l>
       <l>At his heels a stone.</l>
     </lg>
     <p>O, ho!</p>
   </sp>
  </div2>
</div1>
```

Source: [175]

The **next** and **prev** attributes are discussed in section *16.7. Aggregation*: they form part of the module for alignment and linking; this module must therefore be included in a schema if they are to be used, as further discussed in section *1.2. Defining a TEI Schema*.

The fragments of Ophelia's song might also be linked together using the <join> mechanism described in section *16.7. Aggregation*. The <join> element is specifically intended to encode the fact that several discontiguous elements of the text together form one 'virtual' element. Using this mechanism, the example might be encoded as follows:

```
<text>
  <body>
    <div1 n="4" type="act">
      <div2 n="5" type="scene">
        <stage type="setting">Elsinore. A room in the Castle.</stage>
        <sp>
          <speaker>Queen</speaker>
          <p>How now, Ophelia?</p>
        </sp>
        <sp>
          <speaker>Ophelia</speaker>
          <stage type="delivery">Singing</stage>
          <lg xml:id="TL1" type="song" part="Y">
            <l>How should I your true-love know</l>
            <l>From another one?</l>
            <l>By his cockle hat and staff</l>
            <l>And his sandal shoon.</l>
          </lg>
        </sp>
        <sp>
          <speaker>Queen</speaker>
          <p>Alas, sweet lady, what imports this song?</p>
        </sp>
        <sp>
          <speaker>Ophelia</speaker>
          <p>Say you? Nay, pray you mark.</p>
          <stage type="delivery">Sings</stage>
          <lg xml:id="TL2" type="song" part="Y">
            <l>He is dead and gone, lady,</l>
            <l>He is dead and gone;</l>
            <l>At his head a grass-green turf,</l>
```

```
          <l>At his heels a stone.</l>
        </lg>
        <p>O, ho!</p>
        <join  type="lg"  targets="#TL1 #TL2"/>
      </sp>
    </div2>
  </div1>
 </body>
</text>
```

The location of the <join> element is not significant; here it has been placed shortly after the conclusion of the song, in order to have it close to the fragments it unifies.

Like the **next** and **prev** attributes, the <join> element requires the additional module for linking, which is selected as shown above.

7.2.6 Simultaneous Action

In printed or written versions of performance texts, a variety of techniques may be used to indicate the temporal alignment of speeches or actions. Speeches may be printed vertically aligned on the page, or braced together; stage directions (e.g. 'Speaking at the same time') are also often used. In operatic or musical works in particular, the need to indicate timing and alignment of individual parts of a song may lead to very complex layout.

One simple method of indicating the temporal alignment of speeches or actions is to use the **corresp** attribute discussed in section *16.4. Correspondence and Alignment*, as in the following example:

```
<sp>
  <speaker>Mangan</speaker>
  <stage  type="delivery">wildly</stage>
  <p>Look here: I'm going to take off all my clothes.</p>
  <stage  type="action">he begins tearing off his coat.</stage>
</sp>
<sp  xml:id="dr-s1">
  <speaker>Lady Utterword</speaker>
  <p>Mr Mangan!</p>
</sp>
<sp  xml:id="dr-s2">
  <speaker>Captain Shotover</speaker>
  <p>Whats that?</p>
</sp>
<sp  xml:id="dr-s3">
  <speaker>Hector</speaker>
  <p>Ha! ha! Do. Do.</p>
</sp>
<sp  xml:id="dr-s4">
  <speaker>Ellie</speaker>
  <p>Please dont.</p>
</sp>
<stage
    corresp="#dr-s1 #dr-s2 #dr-s3 #dr-s4"
    xml:id="dr-d1"
    rend="braced"
    type="delivery">in consternation</stage>
<sp>
  <speaker>Mrs. Hushabye</speaker>
  <stage  type="action">catching his arm and stopping him</stage>
```

```
  <p>Alfred: for shame! Are you mad?</p>
</sp>
```

Source: [179]

In the original, the stage direction 'in consternation' is printed opposite a brace grouping all four speeches, indicating that all four characters speak at once, and that the stage direction applies to all of them. Rather than attempting to represent the appearance of the source, this example encoding represents its presumed meaning: the <stage> element is placed arbitrarily after the last relevant speech, and the four speeches with which it is to be associated are pointed to by means of the **corresp** attribute. This attribute, which is enabled by the linking module, provides a simple way of indicating the temporal alignment of speeches or actions in a play. Producing a readable version of the text which simulates the original printed effect may however require more complex markup and processing.

More powerful and more precise mechanisms for temporal alignment are defined in chapter *8. Transcriptions of Speech*. These would be appropriate for encodings the focus of which is on the actual performance of a text rather than its structure or formal properties. The module described in that chapter includes a large number of other detailed proposals for the encoding of such features as voice quality, prosody, etc., which might be relevant to such a treatment of performance texts.

7.3 Other Types of Performance Text

Most of the elements and structures identified thus far are derived from traditional theatrical texts. Although other performance texts, such as screenplays or radio scripts, have not been discussed specifically, they can be encoded using the elements and structures listed above. Encoders may however find it convenient to use, as well, the additional specialized elements discussed in this section. For scripts containing very detailed technical information, the <tech> element discussed in section *7.3.1. Technical Information* may also be useful.

Like other texts, screenplays and television or radio scripts may be divided into text divisions marked with or <div1>, etc. Within units corresponding with the traditional 'act' and 'scene', further subdivisions or sequences may be identified, composed of individual 'shots', each associated with a single camera angle and setting. Shots and sequences should be encoded using an appropriate text-division element (i.e., a <div3> element if numbered division elements are in use and the next largest unit is a <div2>, or a element if un-numbered divisions are in use) specifying **sequence** or **shot** as the value of the **type** attribute, as appropriate.

It is normal practice in screenplays and radio scripts to distinguish directions concerning camera angles, sound effects, etc., from other forms of stage direction. Such texts also generally include far more detailed specifications of what the audience actually sees: descriptions of actions and background, etc. Scripts derived from cinema and television productions may also include texts displayed as captions superimposed on the action. All of these may be encoded using the general purpose <stage> element discussed in section *7.2.3. Stage Directions*, and distinguished by means of its **type** attribute. Alternatively, or in addition, the following more specific elements may be used, where clear distinctions can be made:

<view> describes the visual context of some part of a screen play in terms of what the spectator sees, generally independent of any dialogue.

<camera> describes a particular camera angle or viewpoint in a screen play.

<caption> contains the text of a caption or other text displayed as part of a film script or screenplay.

<sound> describes a sound effect or musical sequence specified within a screen play or radio script.

- @**type** categorizes the sound in some respect, e.g. as music, special effect, etc.
- @**discrete** indicates whether the sound overlaps the surrounding speeches or interrupts them.

Some examples of the use of these elements follow:

```
<camera>Angle on Olivia.</camera>
<view>Ryan's wife, standing nervously alone on the sidelines,
biting her lip. She's scared and she shows it.</view>
```

Where particular words or phrases within a direction are emphasized (by change of typeface or use of capital letters), an appropriate phrase-level element may be used to indicate the fact, as in the following examples, where certain words in the original are given in small capitals:

```
<view>George glances at the window--and freezes.
<camera>New angle--shock cut</camera> Out the window
the body of a dead man suddenly slams into
<hi>frame</hi>. He dangles grotesquely,
held up by his coat caught on a protruding bolt.
George gasps. The train <hi>whistle</hi> screams.</view>
```

```
<view>Ext. TV control van—Early morning.
The <name>T.V. announcer</name> from the Ryan interview
stands near the Control Van, the lake in b.g.</view>
<sp>
  <speaker>T.V. Announcer</speaker>
  <p>Several years ago, Jack Ryan was a highly
     successful hydroplane racer ...</p>
</sp>
```

All of these elements, like other stage directions, can appear both within and between speeches.

```
<sp>
  <speaker>TV Announcer VO</speaker>
  <p>Working with Ryan are his two coworkers—
     Strut Bowman, the mechanical engineer—
  <view>
      <camera>Angle on Strut</camera>
         standing in the tow boat, walkie-talkie in hand,
         watching Ryan carefully.</view>
     —and Roger Dalton, a rocket
     systems analyst, and one of the scientists
     from the Jet Propulsion Lab ...</p>
</sp>
```

```
<sp>
  <speaker>Benjy</speaker>
  <p>Now to business.</p>
</sp>
<sp>
  <speaker>Ford and Zaphod</speaker>
  <p>To business.</p>
</sp>
<sound>Glasses clink.</sound>
<sp>
```

```
  <speaker>Benjy</speaker>
  <p>I beg your pardon?</p>
</sp>
<sp>
  <speaker>Ford</speaker>
  <p>I'm sorry, I thought you were proposing a toast.</p>
</sp>
```

Source: [1]

```
<camera>Zoom in to overlay showing some stock film
of hansom cabs galloping past.</camera>
<caption>London, 1895.</caption>
<caption>The residence of Mr Oscar Wilde.</caption>
<sound>Suitably classy music starts.</sound>
<view>Mix through to Wilde's drawing room. A crowd of suitably
dressed folk are engaged in typically brilliant conversation,
laughing affectedly and drinking champagne.</view>
<sp>
  <speaker>Prince of Wales</speaker>
  <p>My congratulations, Wilde. Your latest play is a great success.</p>
</sp>
```

Source: [103]

7.3.1 Technical Information

Traditional stage scripts may contain additional technical information about such production-related factors as lighting, 'blocking' (that is, detailed notes on actors' movements), or props required at particular points. More technical information about intended production effects may also appear in published versions of screenplays or movie scripts. Where these are presented simply as marginal notes, they may be encoded using the general-purpose <note> element defined in section *3.8. Notes, Annotation, and Indexing*. Alternatively, they may be formally distinguished from other stage directions by using the specialized <tech> element:

<tech> (technical stage direction) describes a special-purpose stage direction that is not meant for the actors.

@**type** categorizes the technical stage direction.

@**perf** (performance) identifies the performance or performances to which this technical direction applies.

Like stage directions, <tech> elements can appear anywhere within a speech or between speeches.

7.4 Module for Performance Texts

The module described in this chapter makes available the following components:

Module drama: Performance texts

- Elements defined: actor camera caption castGroup castItem castList epilogue move performance prologue role roleDesc set sound tech view

The selection and combination of modules to form a TEI schema is described in *1.2. Defining a TEI Schema*.

Chapter 8

Transcriptions of Speech

The module described in this chapter is intended for use with a wide variety of transcribed spoken material. It should be stressed, however, that the present proposals are not intended to support unmodified every variety of research undertaken upon spoken material now or in the future; some discourse analysts, some phonologists, and doubtless others may wish to extend the scheme presented here to express more precisely the set of distinctions they wish to draw in their transcriptions. Speech regarded as a purely acoustic phenomenon may well require different methods from those outlined here, as may speech regarded solely as a process of social interaction.

This chapter begins with a discussion of some of the problems commonly encountered in transcribing spoken language (section *8.1. General Considerations and Overview*). Section *8.2. Documenting the Source of Transcribed Speech* documents some additional TEI Header elements which may be used to document the recording or other source from which transcribed text is taken. Section *8.3. Elements Unique to Spoken Texts* describes the basic structural elements provided by this module. Finally, section *8.4. Elements Defined Elsewhere* of this chapter reviews further problems specific to the encoding of spoken language, demonstrating how mechanisms and elements discussed elsewhere in these Guidelines may be applied to them.

8.1 General Considerations and Overview

There is great variation in the ways different researchers have chosen to represent speech using the written medium.[1] This reflects the special difficulties which apply to the encoding or transcription of speech. Speech varies according to a large number of dimensions, many of which have no counterpart in writing (for example, tempo, loudness, pitch, etc.). The audibility of speech recorded in natural communication situations is often less than perfect, affecting the accuracy of the transcription. Spoken material may be transcribed in the course of linguistic, acoustic, anthropological, psychological, ethnographic, journalistic, or many other types of research. Even in the same field, the interests and theoretical perspectives of different transcribers may lead them to prefer different levels of detail in the transcript and different styles of visual display. The production and comprehension of speech are intimately bound up with the situation in which speech occurs, far more so than is the case for written texts. A speech transcript must therefore include some contextual features; determining which are relevant is not always simple. Moreover, the ethical problems in recording and making public what was produced in a private setting and intended for a limited audience are more frequently encountered in dealing with spoken texts than with written ones.

Speech also poses difficult structural problems. Unlike a written text, a speech event takes place in time. Its beginning and end may be hard to determine and its internal composition difficult to define. Most researchers agree that the utterances or turns of individual speakers form an important structural component in most kinds of speech, but these are rarely as well-behaved (in the structural sense) as paragraphs or other analogous

[1] For a discussion of several of these see Edwards and Lampert (eds.) (1993); Johansson (1994); and Johansson et al. (1991).

units in written texts: speakers frequently interrupt each other, use gestures as well as words, leave remarks unfinished and so on. Speech itself, though it may be represented as words, frequently contains items such as vocalized pauses which, although only semi-lexical, have immense importance in the analysis of spoken text. Even non-vocal elements such as gestures may be regarded as forming a component of spoken text for some analytic purposes. Below the level of the individual utterance, speech may be segmented into units defined by phonological, prosodic, or syntactic phenomena; no clear agreement exists, however, even as to appropriate names for such segments.

Spoken texts transcribed according to the guidelines presented here are organized as follows. The overall structure of a TEI spoken text is identical to that of any other TEI text: the <TEI> element for a spoken text contains a <teiHeader> element, followed by a <text> element. Even texts primarily composed of transcribed speech may also include conventional front and back matter, and may even be organized into divisions like printed texts.

We may say, therefore, that these Guidelines regard transcribed speech as being composed of arbitrary high-level units called texts. A spoken <text> might typically be a conversation between a small number of people, a lecture, a broadcast TV item, or a similar event. Each such unit has associated with it a <teiHeader> providing detailed contextual information such as the source of the transcript, the identity of the participants, whether the speech is scripted or spontaneous, the physical and social setting in which the discourse takes place and a range of other aspects. Details of the header in general are provided in chapter *2. The TEI Header*; the particular elements it provides for use with spoken texts are described below (*8.2. Documenting the Source of Transcribed Speech*). Details concerning additional elements which may be used for the documentation of participant and contextual information are given in *15.2. Contextual Information*.

Defining the bounds of a spoken text is frequently a matter of arbitrary convention or convenience. In public or semi-public contexts, a text may be regarded as synonymous with, for example, a lecture, a broadcast item, a meeting, etc. In informal or private contexts, a text may be simply a conversation involving a specific group of participants. Alternatively, researchers may elect to define spoken texts solely in terms of their duration in time or length in words. By default, these Guidelines assume of a text only that:

- it is internally cohesive,
- it is describable by a single header, and
- it represents a single stretch of time with no significant discontinuities.

Deviation from these assumptions may be specified (for example, the **org** attribute on the <text> element may take the value **compos** to specify that the components of the text are discrete) but is not recommended.

Within a <text> it may be necessary to identify subdivisions of various kinds, if only for convenience of handling. The neutral element discussed in section *4.1. Divisions of the Body* is recommended for this purpose. It may be found useful also for representing subdivisions relating to discourse structure, speech act theory, transactional analysis, etc., provided only that these divisions are hierarchically well-behaved. Where they are not, as is often the case, the mechanisms discussed in chapters *16. Linking, Segmentation, and Alignment* and *20. Non-hierarchical Structures* may be used.

A spoken text may contain any of the following components:

- utterances
- pauses
- vocalized but non-lexical phenomena such as coughs
- kinesic (non-verbal, non-lexical) phenomena such as gestures
- entirely non-linguistic incidents occurring during and possibly influencing the course of speech
- writing, regarded as a special class of incident in that it can be transcribed, for example captions or overheads displayed during a lecture

- shifts or changes in vocal quality

Elements to represent all of these features of spoken language are discussed in section *8.3. Elements Unique to Spoken Texts* below.

An utterance (tagged <u>) may contain lexical items interspersed with pauses and non-lexical vocal sounds; during an utterance, non-linguistic incidents may occur and written materials may be presented. The <u> element can thus contain any of the other elements listed, interspersed with a transcription of the lexical items of the utterance; the other elements may all appear between utterances or next to each other, but except for <writing> they do not contain any other elements nor any data.

A spoken text itself may be without substructure, that is, it may consist simply of units such as utterances or pauses, not grouped together in any way, or it may be subdivided. If the notion of what constitutes a 'text' in spoken discourse is inevitably rather an arbitrary one, the notion of formal subdivisions within such a 'text' may appear even more debatable. Nevertheless, such divisions may be useful for such types of discourse as debates, broadcasts, etc., where structural subdivisions can easily be identified, or more generally wherever it is desired to aggregate utterances or other parts of a transcript into units smaller than a complete 'text'. Examples might include 'conversations' or 'discourse fragments', or more narrowly, 'that part of the conversation where topic x was discussed', provided only that the set of all such divisions is coextensive with the text.

Each such division of a spoken text should be represented by the numbered or un-numbered elements defined in chapter *4. Default Text Structure*. For some detailed kinds of analysis a hierarchy of such divisions may be found useful; nested elements may be used for this purpose, as in the following example showing how a collection made up of transcribed 'sound bites' taken from speeches given by a politician on different occasions, might be encoded. Each extract is regarded as a distinct , nested within a single composite as follows:

```
<div type="soundbites" subtype="conservative" org="composite">
  <div sample="medial"/>
  <div sample="medial"/>
  <div sample="initial"/>
</div>
```

As a member of the class att.declaring, the element may also carry a **decls** attribute, for use where the divisions of a text do not all share the same set of the contextual declarations specified in the TEI header. (See further section *15.3. Associating Contextual Information with a Text*).

8.2 Documenting the Source of Transcribed Speech

Where a computer file is derived from a spoken text rather than a written one, it will usually be desirable to record additional information about the recording or broadcast which constitutes its source. Several additional elements are provided for this purpose within the source description component of the TEI Header:

<scriptStmt> (script statement) contains a citation giving details of the script used for a spoken text.

<recordingStmt> (recording statement) describes a set of recordings used as the basis for transcription of a spoken text.

<recording> (recording event) details of an audio or video recording event used as the source of a spoken text, either directly or from a public broadcast.

@**type** the kind of recording.

As a member of the att.duration class, the <recording> element inherits the following attribute:

att.duration.w3c attributes for recording normalized temporal durations.

@**dur** (duration) indicates the length of this element in time.

Note that detailed information about the participants or setting of an interview or other transcript of spoken language should be recorded in the appropriate division of the profile description, discussed in chapter *15. Language Corpora*, rather than as part of the source description. The source description is used to hold information only about the source from which the transcribed speech was taken, for example, any script being read and any technical details of how the recording was produced. If the source was a previously-created transcript, it should be treated in the same way as any other source text.

The <scriptStmt> element should be used where it is known that one or more of the participants in a spoken text is speaking from a previously prepared script. The script itself should be documented in the same way as any other written text, using one of the three citation tags mentioned above. Utterances or groups of utterances may be linked to the script concerned by means of the **decls** attribute, described in section *15.3. Associating Contextual Information with a Text*.

```
<sourceDesc>
  <scriptStmt  xml:id="CNN12">
    <bibl>
      <author>CNN Network News</author>
      <title>News headlines</title>
      <date  when="1991-06-12">12 Jun 91</date>
    </bibl>
  </scriptStmt>
</sourceDesc>
```

The <recordingStmt> is used to group together information relating to the recordings from which the spoken text was transcribed. The element may contain either a prose description or, more helpfully, one or more <recording> elements, each corresponding with a particular recording. The linkage between utterances or groups of utterances and the relevant recording statement is made by means of the **decls** attribute, described in section *15.3. Associating Contextual Information with a Text*.

The <recording> element should be used to provide a description of how and by whom a recording was made. This information may be provided in the form of a prose description, within which such items as statements of responsibility, names, places, and dates may be identified using the appropriate phrase-level tags. Alternatively, a selection of elements from the model.recordingPart class may be provided. This element class makes available the following elements:

<date> contains a date in any format.

<time> contains a phrase defining a time of day in any format.

<respStmt> (statement of responsibility) supplies a statement of responsibility for the intellectual content of a text, edition, recording, or series, where the specialized elements for authors, editors, etc. do not suffice or do not apply.

<equipment> provides technical details of the equipment and media used for an audio or video recording used as the source for a spoken text.

<broadcast> describes a broadcast used as the source of a spoken text.

Specialized collections may wish to add further sub-elements to these major components. These elements should be used only for information relating to the recording process itself; information about the setting or participants (for example) is recorded elsewhere: see sections *15.2.3. The Setting Description* and *15.2.2. The Participant Description* below.

```
<recordingStmt>
  <recording  type="video">
    <p>U-matic recording made by college audio-visual department staff,
```

```
       available as PAL-standard VHS transfer or sound-only casssette</p>
  </recording>
</recordingStmt>
```

```
<recordingStmt>
  <recording  type="audio"  dur="P30M">
    <respStmt>
      <resp>Location recording by</resp>
      <name>Sound Services Ltd.</name>
    </respStmt>
    <equipment>
      <p>Multiple close microphones mixed down to stereo Digital
            Audio Tape, standard play, 44.1 KHz sampling frequency</p>
    </equipment>
    <date>12 Jan 1987</date>
  </recording>
</recordingStmt>
```

```
<recordingStmt>
  <recording  type="audio"  dur="P15M"  xml:id="rec-3001">
    <date>14 Feb 2001</date>
  </recording>
  <recording  type="audio"  dur="P15M"  xml:id="rec-3002">
    <date>17 Feb 2001</date>
  </recording>
  <recording  type="audio"  dur="P15M"  xml:id="rec-3003">
    <date>22 Feb 2001</date>
  </recording>
</recordingStmt>
```

When a recording has been made from a public broadcast, details of the broadcast itself should be supplied within the <recording> element, as a nested <broadcast> element. A broadcast is closely analogous to a publication and the <broadcast> element should therefore contain one or the other of the bibliographic citation elements <bibl>, <biblStruct>, or <biblFull>. The broadcasting agency responsible for a broadcast is regarded as its author, while other participants (for example interviewers, interviewees, script writers, directors, producers, etc.) should be specified using the <respStmt> or <editor> element with an appropriate <resp> (see further section *3.11. Bibliographic Citations and References*).

```
<recording  type="audio"  dur="P10M">
  <equipment>
    <p>Recorded from FM Radio to digital tape</p>
  </equipment>
  <broadcast>
    <bibl>
      <title>Interview on foreign policy</title>
      <author>BBC Radio 5</author>
      <respStmt>
        <resp>interviewer</resp>
        <name>Robin Day</name>
      </respStmt>
```

```
    <respStmt>
      <resp>interviewee</resp>
      <name>Margaret Thatcher</name>
    </respStmt>
    <series>
      <title>The World Tonight</title>
    </series>
    <note>First broadcast on <date  when="1989-11-27">27 Nov 1989</date>
    </note>
   </bibl>
  </broadcast>
</recording>
```

When a broadcast contains several distinct recordings (for example a compilation), additional <recording> elements may be further nested within the <broadcast> element.

```
<recording  dur="P100M">
  <broadcast>
    <recording/>
  </broadcast>
</recording>
```

8.3 Elements Unique to Spoken Texts

The following elements characterize spoken texts, transcribed according to these Guidelines:

<u> (utterance) a stretch of speech usually preceded and followed by silence or by a change of speaker.

<pause/> a pause either between or within utterances.

<vocal> any vocalized but not necessarily lexical phenomenon, for example voiced pauses, non-lexical backchannels, etc.

<kinesic> any communicative phenomenon, not necessarily vocalized, for example a gesture, frown, etc.

<incident> any phenomenon or occurrence, not necessarily vocalized or communicative, for example incidental noises or other events affecting communication.

<writing> a passage of written text revealed to participants in the course of a spoken text.

<shift/> marks the point at which some paralinguistic feature of a series of utterances by any one speaker changes.

The <u> element may appear directly within a spoken text, and may contain any of the others; the others may also appear directly (for example, a vocal may appear between two utterances) but cannot contain a <u> element. In terms of the basic TEI model, therefore, we regard the <u> element as analogous to a paragraph, and the others as analogous to 'phrase' elements, but with the important difference that they can exist either as siblings or as children of utterances. The class model.divPart.spoken provides the <u> element; the class model.phrase.spoken provides the six other elements listed above.

As members of the att.ascribed class, all of these elements share the following attribute:

att.ascribed provides attributes for elements representing speech or action that can be ascribed to a specific individual.

@who indicates the person, or group of people, to whom the element content is ascribed.

As members of the att.typed and att.timed and att.duration classes, all of these elements except <shift> share the following attribute:

att.typed provides attributes which can be used to classify or subclassify elements in any way.

@**type** characterizes the element in some sense, using any convenient classification scheme or typology.

@**subtype** provides a sub-categorization of the element, if needed

att.timed provides attributes common to those elements which have a duration in time, expressed either absolutely or by reference to an alignment map.

@**start** indicates the location within a temporal alignment at which this element begins.

@**end** indicates the location within a temporal alignment at which this element ends.

att.duration.w3c attributes for recording normalized temporal durations.

@**dur** (duration) indicates the length of this element in time.

Each of these elements is further discussed and specified below in sections *8.3.1. Utterances* to *8.3.4. Writing*.

We can show the relationship between four of these constituents of speech using the features *eventive*, *communicative*, *anthropophonic* (for sounds produced by the human vocal apparatus), and *lexical*:

	eventive	communicative	anthropophonic	lexical
incident	+	-	-	-
kinesic	+	+	-	-
vocal	+	+	+	-
utterance	+	+	+	+

The differences are not always clear-cut. Among incidents might be included actions like slamming the door, which can certainly be communicative. Vocals include coughing and sneezing, which are usually involuntary noises. Equally, the distinction between utterances and vocals is not always clear, although for many analytic purposes it will be convenient to regard them as distinct. Individual scholars may differ in the way borderlines are drawn and should declare their definitions in the <editorialDecl> element of the header (see *2.3.3. The Editorial Practices Declaration*).

The following short extract exemplifies several of these elements. It is recoded from a text originally transcribed in the CHILDES format.[2] Each utterance is encoded using a <u> element (see section *8.3.1. Utterances*). The speakers are defined using the <listPerson> element discussed in *13.3.2. The Person Element* and each is given a unique identifier also used to identify their speech. Pauses marked by the transcriber are indicated using the <pause> element (see section *8.3.2. Pausing*). Non-verbal vocal effects such as the child's meowing are indicated either with orthographic transcriptions or with the <vocal> element, and entirely non-linguistic but significant incidents such as the sound of the toy cat are represented by the <incident> elements (see section *8.3.3. Vocal, Kinesic, Incident*).

```
<u  who="#mar">you
never <pause/> take this cat for show and tell
<pause/> meow meow</u>
<u  who="#ros">yeah well I dont want to</u>
<incident>
  <desc>toy cat has bell in tail which continues to make a tinkling sound</desc>
</incident>
<vocal  who="#mar">
  <desc>meows</desc>
</vocal>
<u  who="#ros">because it is so old</u>
```

[2] The original is a conversation between two children and their parents, recorded in 1987, and discussed in MacWhinney (1988)

```
<u  who="#mar">how <choice>
    <orig>bout</orig>
    <reg>about</reg>
  </choice>
  <emph>your</emph> cat <pause/>yours is <emph>new</emph>
  <kinesic>
    <desc>shows Father the cat</desc>
  </kinesic>
</u>
<u  trans="pause"  who="#fat">thats <pause/> darling</u>
<u  who="#mar">
  <seg>no <emph>mine</emph> isnt old</seg>
  <seg>mine is just um a little dirty</seg>
</u>
<!-- ... -->
<listPerson>
  <person  xml:id="mar">
<!-- ... -->
  </person>
  <person  xml:id="ros">
<!-- ... -->
  </person>
  <person  xml:id="fat">
<!-- ... -->
  </person>
</listPerson>
```

Source: [134]

This example also uses some elements common to all TEI texts, notably the <reg> tag for editorial regularization. Unusually stressed syllables have been encoded with the <emph> element. The <seg> element has also been used to segment the last utterance. Further discussion of all of such options is provided in section *8.4. Elements Defined Elsewhere*.

Contextual information is of particular importance in spoken texts, and should be provided by the TEI header of a text. In general, all of the information in a header is understood to be relevant to the whole of the associated text. The element <u> as a member of the att.declaring class, may however specify a different context by means of the **decls** attribute (see further section *15.3. Associating Contextual Information with a Text*).

8.3.1 Utterances

Each distinct utterance in a spoken text is represented by a <u> element, described as follows:

<u> (utterance) a stretch of speech usually preceded and followed by silence or by a change of speaker.

@**trans** (transition) indicates the nature of the transition between this utterance and the previous one.

Use of the **who** attribute to associate the utterance with a particular speaker is recommended but not required. Its use implies as a further requirement that all speakers be identified by a <person> or <personGrp> element in the TEI header (see section *15.2.2. The Participant Description*). Where utterances or other parts of the transcription cannot be attributed with confidence to any particular participant or group of participants, the encoder may choose to define 'participants' such as *all* or *various*, or *unknown*.

The **trans** attribute is provided as a means of characterizing the transition from one utterance to the next at a simpler level of detail than that provided by the temporal alignment mechanism discussed in section *16.5.*

Synchronization. The value specified applies to the transition from the preceding utterance into the utterance bearing the attribute. For example:[3]

```
<u xml:id="ts_a1" who="#a">Have you heard the</u>
<u xml:id="ts_b1" trans="latching" who="#b">the election results? yes</u>
<u xml:id="ts_a2" trans="pause" who="#a">it's a disaster</u>
<u xml:id="ts_b2" trans="overlap" who="#b">it's a miracle</u>
```

In this example, utterance **ts_b1** latches on to utterance **ts_a1**, while there is a marked pause between **ts_b1** and **ts_a2**. **ts_b2** and **ts_a2** overlap, but by an unspecified amount. For ways of providing a more precise indication of the degree of overlap, see section *8.4.2. Synchronization and Overlap*.

An utterance may contain either running text, or text within which other basic structural elements are nested. Where such nesting occurs, the **who** attribute is considered to be inherited for the elements <pause>, <vocal>, <shift> and <kinesic>; that is, a pause or shift (etc.) within an utterance is regarded as being produced by that speaker only, while a pause between utterances applies to all speakers.

Occasionally, an utterance may seem to contain other utterances, for example where one speaker interrupts himself, or when another speaker produces a 'back-channel' while they are still speaking. The present version of these Guidelines does not support nesting of one <u> element within another. The transcriber must therefore decide whether such interruptions constitute a change of utterance, or whether other elements may be used. In the case of self-interruption, the <shift> element may be used to show that the speaker has changed the quality of their speech:

```
<u who="#a">Listen to this <shift new="reading"/>The government is
confident, he said, that the current economic problems will be
completely overcome by June<shift/> what nonsense</u>
```

Alternatively the <incident> element described in section *8.3.3. Vocal, Kinesic, Incident* might be used, without transcribing the read material:

```
<u who="#a">Listen to this
<incident>
   <desc>reads aloud from newspaper</desc>
  </incident> what
nonsense</u>
```

Often, back-channelling is only semi-lexicalized and may therefore be represented using the <vocal> element:

```
<u who="#a">So what could I have done <vocal who="#b">
   <desc>tut-tutting</desc>
  </vocal> about it anyway?</u>
```

Where this is not possible, it is simplest to regard the back-channel as a distinct utterance.

[3] For the most part, the examples in this chapter use no sentence punctuation except to mark the rising intonation often found in interrogative statements; for further discussion, see section *8.4.3. Regularization of Word Forms*.

8.3.2 Pausing

Speakers differ very much in their rhythm and in particular in the amount of time they leave between words. The following element is provided to mark occasions where the transcriber judges that speech has been paused, irrespective of the actual amount of silence:

<pause/> a pause either between or within utterances.

A pause contained by an utterance applies to the speaker of that utterance. A pause between utterances applies to all speakers. The **type** attribute may be used to categorize the pause, for example as short, medium, or long; alternatively the attribute **dur** may be used to indicate its length more exactly, as in the following example:

```
<u>Okay <pause dur="PT2M"/>U-m<pause dur="PT75S"/>the scene opens up
<pause dur="PT50S"/> with <pause dur="PT20S"/> um <pause dur="PT145S"/> you see
a tree okay?</u>
```

Source: [32]

If detailed synchronization of pausing with other vocal phenomena is required, the alignment mechanism defined at section *16.5. Synchronization* and discussed informally below should be used. Note that the **trans** attribute mentioned in the previous section may also be used to characterize the degree of pausing between (but not within) utterances.

8.3.3 Vocal, Kinesic, Incident

These three empty elements are used to indicate the presence of non-transcribed semi-lexical or non-lexical phenomena either between or within utterances.

<vocal> any vocalized but not necessarily lexical phenomenon, for example voiced pauses, non-lexical backchannels, etc.

<kinesic> any communicative phenomenon, not necessarily vocalized, for example a gesture, frown, etc.

<incident> any phenomenon or occurrence, not necessarily vocalized or communicative, for example incidental noises or other events affecting communication.

The **who** attribute should be used to specify the person or group responsible for a vocal, kinesic, or incident which is contained within an utterance, if this differs from that of the enclosing utterance. The attribute must be supplied for a vocal, kinesic, or incident which is not contained within an utterance.

The **iterated** attribute may be used to indicate that the vocal, kinesic, or incident is repeated, for example **laughter** as opposed to **laugh**. These should both be distinguished from **laughing**, where what is being encoded is a shift in voice quality. For this last case, the <shift> element discussed in section *8.3.6. Shifts* should be used.

A child <desc> element may be used to supply a conventional representation for the phenomenon, for example:

non-lexical burp, click, cough, exhale, giggle, gulp, inhale, laugh, sneeze, sniff, snort, sob, swallow, throat, yawn

semi-lexical ah, aha, aw, eh, ehm, er, erm, hmm, huh, mm, mmhm, oh, ooh, oops, phew, tsk, uh, uh-huh, uh-uh, um, urgh, yup

Researchers may prefer to regard some semi-lexical phenomena as 'words' within the bounds of the <u> element. See further the discussion at section *8.4.3. Regularization of Word Forms* below. As for all basic categories, the definition should be made clear in the <encodingDesc> element of the TEI header.

Some typical examples follow:

```
<u  who="#jan">This is just delicious</u>
<incident>
  <desc>telephone rings</desc>
</incident>
<u  who="#ann">I'll get it</u>
<u  who="#tom">I used to <vocal>
    <desc>cough</desc>
  </vocal> smoke a lot</u>
<u  who="#bob">
  <vocal>
    <desc>sniffs</desc>
  </vocal>He thinks he's tough
</u>
<vocal  who="#ann">
  <desc>snorts</desc>
</vocal>
<!-- ... -->
<listPerson>
  <person  xml:id="ann">
<!-- ... -->
  </person>
  <person  xml:id="bob">
<!-- ... -->
  </person>
  <person  xml:id="jan">
<!-- ... -->
  </person>
  <person  xml:id="kim">
<!-- ... -->
  </person>
  <person  xml:id="tom">
<!-- ... -->
  </person>
</listPerson>
```

Source: [6]

Note that Ann's snorting could equally well be encoded as follows:

```
<u  who="#ann">
  <vocal>
    <desc>snorts</desc>
  </vocal>
</u>
```

The extent to which encoding of incidents or kinesics is included in a transcription will depend entirely on the purpose for which the transcription was made. As elsewhere, this will depend on the particular research agenda and the extent to which their presence is felt to be significant for the interpretation of spoken interactions.

8.3.4 Writing

Written text may also be encountered when speech is transcribed, for example in a television broadcast or cinema performance, or where one participant shows written text to another. The <writing> element may be used to distinguish such written elements from the spoken text in which they are embedded.

<writing> a passage of written text revealed to participants in the course of a spoken text.
- @**gradual** indicates whether the writing is revealed all at once or gradually.
- @**source** points to a bibliographic citation in the header giving a full description of the source or script of the writing.

For example, if speaker A in the breakfast table conversation in section *8.3.1. Utterances* above had simply shown the newspaper passage to her interlocutor instead of reading it, the interaction might have been encoded as follows:

```
<u  who="#a">look at this</u>
<writing  who="#a"  type="newspaper"  gradual="false">Government claims economic problems
<soCalled>over by June</soCalled>
</writing>
<u  who="#a">what nonsense!</u>
```

If the source of the writing being displayed is known, bibliographic information about it may be stored in a <listBibl> within the <sourceDesc> element of the TEI Header, and then pointed to using the **source** attribute. For example, in the following imaginary example, a lecturer displays two different versions of the same passage of text:

```
<sourceDesc>
<!-- ...-->
  <bibl  xml:id="FOL1">Shakespeare First Folio text</bibl>
  <bibl  xml:id="FOL2">Shakespeare Second Folio text</bibl>
<!-- ...-->
</sourceDesc>
<!-- ...-->
<u>.... now compare the punctuation of lines 12 and 14 in these two
versions of page 42...
<writing  source="#FOL1">....</writing>
  <writing  source="#FOL2">....</writing>
</u>
```

8.3.5 Temporal Information

As noted above, utterances, vocals, pauses, kinesics, incidents, and writing elements all inherit attributes providing information about their position in time from the classes att.timed and att.duration. These attributes can be used to link parts of the transcription very exactly with points on a timeline, or simply to indicate their duration. Note that if **start** and **end** point to <when> elements whose temporal distance from each other is specified in a timeline, then **dur** is ignored.

The <anchor> element (see *16.4. Correspondence and Alignment*) may be used as an alternative means of aligning the start and end of timed elements, and is required when the temporal alignment involves points within an element.

For further discussion of temporal alignment and synchronization see *8.4.2. Synchronization and Overlap* below.

8.3.6 Shifts

A common requirement in transcribing spoken language is to mark positions at which a variety of prosodic features change. Many paralinguistic features (pitch, prominence, loudness, etc.) characterize stretches of speech which are not co-extensive with utterances or any of the other units discussed so far. One simple method of encoding such units is simply to mark their boundaries. An empty element called <shift> is provided for this purpose.

<shift/> marks the point at which some paralinguistic feature of a series of utterances by any one speaker changes.

@**feature** a paralinguistic feature.

@**new** specifies the new state of the paralinguistic feature specified.

A <shift> element may appear within an utterance or a segment to mark a significant change in the particular feature defined by its attributes, which is then understood to apply to all subsequent utterances for the same speaker, unless changed by a new shift for the same feature in the same speaker. Intervening utterances by other speakers do not normally carry the same feature. For example:

```
<u>
  <shift feature="loud" new="f"/>Elizabeth
</u>
<u>Yes</u>
<u>
  <shift feature="loud" new="normal"/>Come and try this <pause/>
  <shift feature="loud" new="ff"/>come on
</u>
```

In this example, the word *Elizabeth* is spoken loudly, the words *Yes* and *Come and try this* with normal volume, and the words *come on* very loudly.

The values proposed here for the **feature** attribute are based on those used by the Survey of English Usage (see further Boase 1990); this list may be revised or supplemented using the methods outlined in section *23.2. Personalization and Customization*.

The **new** attribute specifies the new state of the feature following the shift. If no value is specified, it is implied that the feature concerned ceases to be remarkable at this point: the special value **normal** may be specified to have the same effect.

A list of suggested values for each of the features proposed follows:

- tempo

  a allegro (fast)

  aa very fast

  acc accelerando (getting faster)

  l lento (slow)

  ll very slow

  rall rallentando (getting slower)

- loud (for loudness):

  f forte (loud)

  ff very loud

  cresc crescendo (getting louder)

  p piano (soft)

  pp very soft

  dimin diminuendo (getting softer)

- pitch (for pitch range):

high high pitch-range
low low pitch-range
wide wide pitch-range
narrow narrow pitch-range
asc ascending
desc descending
monot monotonous
scand scandent, each succeeding syllable higher than the last, generally ending in a falling tone

- tension:

sl slurred
lax lax, a little slurred
ten tense
pr very precise
st staccato, every stressed syllable being doubly stressed
leg legato, every syllable receiving more or less equal stress

- rhythm:

rh beatable rhythm
arrh arrhythmic, particularly halting
spr spiky rising, with markedly higher unstressed syllables
spf spiky falling, with markedly lower unstressed syllables
glr glissando rising, like spiky rising but the unstressed syllables, usually several, also rise in pitch relative to each other
glf glissando falling, like spiky falling but with the unstressed syllables also falling in pitch relative to each other

- voice (for voice quality):

whisp whisper
breath breathy
husk husky
creak creaky
fals falsetto
reson resonant
giggle unvoiced laugh or giggle
laugh voiced laugh
trem tremulous
sob sobbing
yawn yawning
sigh sighing

A full definition of the sense of the values provided for each feature should be provided in the encoding description section of the text header (see section *2.3. The Encoding Description*).

8.4 Elements Defined Elsewhere

This section describes the following features characteristic of spoken texts for which elements are defined elsewhere in these Guidelines:

- segmentation below the utterance level
- synchronization and overlap
- regularization of orthography

The elements discussed here are not provided by the module for spoken texts. Some of them are included in the core module and others are contained in the modules for linking and for analysis respectively. The selection of modules and their combination to define a TEI schema is discussed in section *1.2. Defining a TEI Schema*.

8.4.1 Segmentation

For some analytic purposes it may be desirable to subdivide the divisions of a spoken text into units smaller than the individual utterance or turn. Segmentation may be performed for a number of different purposes and in terms of a variety of speech phenomena. Common examples include units defined both prosodically (by intonation, pausing, etc.) and syntactically (clauses, phrases, etc.) The term macrosyntagm has been used by a number of researchers to define units peculiar to speech transcripts.[4]

These Guidelines propose that such analyses be performed in terms of neutrally-named segments, represented by the <seg> element, which is discussed more fully in section *16.3. Blocks, Segments, and Anchors*. This element may take a **type** attribute to specify the kind of segmentation applicable to a particular segment, if more than one is possible in a text. A full definition of the segmentation scheme or schemes used should be provided in the <segmentation> element of the <editorialDecl> element in the TEI header (see *2.3.3. The Editorial Practices Declaration*).

In the first example below, an utterance has been segmented according to a notion of syntactic completeness not necessarily marked by the speech, although in this case a pause has been recorded between the two sentence-like units. In the second, the segments are defined prosodically (an acute accent has been used to mark the position immediately following the syllable bearing the primary accent or stress), and may be thought of as 'tone units'.

```
<u>
  <seg>we went to the pub yesterday</seg>
  <pause/>
  <seg>there was no one there</seg>
</u>
<u>
  <seg>although its an old ide´a</seg>
  <seg>it hasnt been on the mar´ket very long</seg>
</u>
```

Source: [152]

In either case, the <segmentation> element in the header of the text should specify the principles adopted to define the segments marked in this way.

When utterances are segmented end-to-end in the same way as the s-units in written texts, the <s> element discussed in chapter *17. Simple Analytic Mechanisms* may be used, either as an alternative or in addition to the more general purpose <seg> element. The <s> element is available without formality in all texts, but does not allow segments to nest within each other.

[4] The term was apparently first proposed by Loman and Jørgensen (1971), where it is defined as follows: 'A text can be analysed as a sequence of segments which are internally connected by a network of syntactic relations and externally delimited by the absence of such relations with respect to neighbouring segments. Such a segment is a syntactic unit called a macrosyntagm' (trans. S. Johansson).

Where segments of different kinds are to be distinguished within the same stretch of speech, the **type** attribute may be used, as in the following example:

```
<u  who="#T1"
   xmlns:ext="http://www.example.org/ns/nonTEI">
  <seg  type="C">I think </seg>
  <seg  type="C">this chap was writing </seg>
  <seg  type="C">and he <del  type="repeated">said hello</del> said </seg>
  <seg  type="M">hello </seg>
  <seg  type="C">and he said </seg>
  <seg  type="C">I'm going to a
  <paraphasia xmlns="http://www.example.org/ns/nonTEI"
    >gate</paraphasia>
     at twenty past seven </seg>
  <seg  type="C">he said </seg>
  <seg  type="M">ok </seg>
  <seg  type="M">right away </seg>
  <seg  type="C">and so <gap  extent="1"/> on they went </seg>
  <seg  type="C">and they were <gap  extent="3"/>
     writing there </seg>
</u>
```

In this example, recoded from a corpus of language-impaired speech prepared by Fletcher and Garman, the speaker's utterance has been fully segmented into clausal (`type="C"`) or minor (`type="M"`) units. An additional element, <ext:paraphasia> has been used to define a particular characteristic of this corpus for which no element exists in the TEI scheme. See further chapter *23.2. Personalization and Customization* for a discussion of the way in which this kind of user-defined extension of the TEI scheme may be performed and chapter *1. The TEI Infrastructure* for the mechanisms on which it depends.

This example also uses the core elements <gap> and <del> to mark editorial decisions concerning matter completely omitted from the transcript (because of inaudibility), and words which have been transcribed but which the transcriber wishes to exclude from the segment because they are repeated, respectively. See section *3.4. Simple Editorial Changes* for a discussion of these and related elements.

It is often the case that the desired segmentation does not respect utterance boundaries; for example, syntactic units may cross utterance boundaries. For a detailed discussion of this problem, and the various methods proposed by these Guidelines for handling it, see chapter *20. Non-hierarchical Structures*. Methods discussed there include these:

- 'milestone' tags may be used; the special-purpose <shift> tag discussed in section *8.3.6. Shifts* is an extension of this method
- where several discontinuous segments are to be grouped together to form a syntactic unit (e.g. a phrasal verb with interposed complement), the <join> element may be used

8.4.2 Synchronization and Overlap

A major difference between spoken and written texts is the importance of the temporal dimension to the former. As a very simple example, consider the following, first as it might be represented in a playscript:

```
Jane: Have you read Vanity Fair?
Stig: Yes
Lou: (nods vigorously)
```

To encode this, we first define the participants:

```
<listPerson>
  <person xml:id="stig">
<!-- ... -->
  </person>
  <person xml:id="lou">
<!-- ... -->
  </person>
  <person xml:id="jane">
<!-- ... -->
  </person>
</listPerson>
```

Let us assume that Stig and Lou respond to Jane's question before she has finished asking it — a fairly normal situation in spontaneous speech. The simplest way of representing this overlap would be to use the **trans** attribute previously discussed:

```
<u who="#jane">have you read Vanity Fair</u>
<u trans="overlap" who="#stig">yes</u>
```

However, this does not allow us to indicate either the extent to which Stig's utterance is overlapped, nor does it show that there are in fact three things which are synchronous: the end of Jane's utterance, Stig's whole utterance, and Lou's kinesic. To overcome these problems, more sophisticated techniques, employing the mechanisms for pointing and alignment discussed in detail in section *16.5. Synchronization*, are needed. If the module for linking has been enabled (as described in section *8.4.1. Segmentation* above), one way to represent the simple example above would be as follows:

```
<u xml:id="utt1" who="#jane">have you read Vanity <anchor synch="#utt2 #k1" xml:id="a1"/> Fair</u>
<u xml:id="utt2" who="#stig">yes</u>
<kinesic xml:id="k1" who="#lou" iterated="true">
  <desc>nods head vertically</desc>
</kinesic>
```

For a full discussion of this and related mechanisms, section *16.5.2. Placing Synchronous Events in Time* should be consulted. The rest of the present section, which should be read in conjunction with that more detailed discussion, presents a number of ways in which these mechanisms may be applied to the specific problem of representing temporal alignment, synchrony, or overlap in transcribing spoken texts.

In the simple example above, the first utterance (that with identifier **utt1**) contains an <anchor> element, the function of which is simply to mark a point within it. The **synch** attribute associated with this anchor point specifies the identifiers of the other two elements which are to be synchronized with it: specifically, the second utterance (**utt2**) and the kinesic (k1). Note that one of these elements has content and the other is empty.

This example demonstrates only a way of indicating a point within one utterance at which it can be synchronized with another utterance and a kinesic. For more complex kinds of alignment, involving possibly multiple synchronization points, an additional element is provided, known as a <timeline>. This consists of a series of <when> elements, each representing a point in time, and bearing attributes which indicate its exact temporal position relative to other elements in the same timeline, in addition to the sequencing implied by its position within it.

For example:

```
<timeline unit="s" origin="#TS-P1">
  <when xml:id="TS-P1" absolute="12:20:01"/>
  <when xml:id="TS-P2" interval="4.5" since="#TS-P1"/>
  <when xml:id="TS-P6"/>
  <when xml:id="TS-P3" interval="1.5" since="#TS-P6"/>
</timeline>
```

This timeline represents four points in time, named TS-P1, TS-P2, TS-P6, and TS-P3 (as with all attributes named **xml:id** in the TEI scheme, the names must be unique within the document but have no other significance). TS-P1 is located absolutely, at 12:20:01:01 BST. TS-P2 is 4.5 seconds later than TS-P2 (i.e. at 12:20:46). TS-P6 is at some unspecified time later than TS-P2 and previous to TS-P3 (this is implied by its position within the timeline, as no attribute values have been specified for it). The fourth point, TS-P3, is 1.5 seconds later than TS-P6.

One or more such timelines may be specified within a spoken text, to suit the encoder's convenience. If more than one is supplied, the **origin** attribute may be used on each to specify which other <timeline> element it follows. The **unit** attribute indicates the units used for timings given on <when> elements contained by the alignment map. Alternatively, to avoid the need to specify times explicitly, the **interval** attribute may be used to indicate that all the <when> elements in a time line are a fixed distance apart.

Three methods are available for aligning points or elements within a spoken text with the points in time defined by the <timeline>:

- The elements to be synchronized may specify the identifier of a <when> element as the value of one of the **start**, **end**, or **synch** attributes
- The <when> element may specify the identifiers of all the elements to be synchronized with it using the **synch** attribute
- A free-standing <link> element may be used to associate the <when> element and the elements synchronized with it by specifying their identifiers as values for its **target** attribute.

  For example, using the timeline given above:

```
<u xml:id="TS-U1" start="#TS-P2" end="#TS-P3">This is my <anchor synch="#TS-P6" xml:id="TS-P6A"/> turn</u>
```

The start of utterance **TS-U1** is aligned with **TS-P2** and its end with **TS-P3**. The transition between the words *my* and *turn* occurs at point **TS-P6A**, which is synchronous with point **TS-P6** on the timeline.

The synchronization represented by the preceding examples could equally well be represented as follows:

```
<timeline origin="#ts-p1" unit="s">
  <when xml:id="ts-p1" absolute="12:20:01"/>
  <when
     synch="#ts-u1"
     xml:id="ts-p2"
     interval="4.5"
     since="#ts-p1"/>
  <when synch="#ts-x1" xml:id="ts-p6"/>
  <when
     synch="#ts-u1"
     xml:id="ts-p3"
     interval="1.5"
     since="#ts-p6"/>
</timeline>
<u xml:id="ts-u1">This is my <anchor xml:id="ts-x1"/> turn</u>
```

Here, the whole of the object with identifier **ts-u1** (the utterance) has been aligned with two different points, **ts-p2** and **ts-p3**. This is interpreted to mean that the utterance spans at least those two points.

Finally, a <linkGrp> may be used as an alternative to the **synch** attribute:

```
<timeline origin="#TS-p1" unit="s">
  <when xml:id="TS-p1" absolute="12:20:01"/>
  <when xml:id="TS-p2" interval="4.5" since="#TS-p1"/>
  <when xml:id="TS-p6"/>
  <when xml:id="TS-p3" interval="1.5" since="#TS-p6"/>
</timeline>
<u xml:id="TS-u1">
  <anchor xml:id="TS-u1start"/>
This is my <anchor xml:id="TS-x1"/> turn
<anchor xml:id="TS-u1end"/>
</u>
<linkGrp type="synchronous">
  <link targets="#TS-u1start #TS-p1"/>
  <link targets="#TS-u1end #TS-p2"/>
  <link targets="#TS-x1 #TS-p6"/>
</linkGrp>
```

As a further example of the three possibilities, consider the following dialogue, represented first as it might appear in a conventional playscript:

```
Tom: I used to smoke - -
Bob: (interrupting) You used to smoke?
Tom: (at the same time) a lot more than this. But I never
inhaled the smoke
```

Source: [6]

A commonly used convention might be to transcribe such a passage as follows:

```
(1) I used to smoke [ a lot more than this ]
(2) [ you used to smoke ]
(1) but I never inhaled the smoke
```

Such conventions have the drawback that they are hard to generalize or to extend beyond the very simple case presented here. Their reliance on the accidentals of physical layout may also make them difficult to transport and to process computationally. These Guidelines recommend the following mechanisms to encode this.

Where the whole of one or another utterance is to be synchronized, the **start** and **end** attributes may be used:

```
<u who="#tom">I used to smoke <anchor xml:id="TS-p10"/> a lot more than this
<anchor xml:id="TS-p20"/>but I never inhaled the smoke</u>
<u start="#TS-p10" end="#TS-p20" who="#bob">You used to smoke</u>
```

Note that the second utterance above could equally well be encoded as follows with exactly the same effect:

```
<u who="#bob">
  <anchor synch="#TS-p10"/>You used to smoke<anchor synch="#TS-p20"/>
</u>
```

If synchronization with specific timing information is required, a <timeline> must be included:

```
<timeline origin="#TS-t01">
  <when xml:id="TS-t01"/>
  <when xml:id="TS-t02"/>
</timeline>
<u who="#tom">I used to smoke
<anchor synch="#TS-t01"/>a lot more than this
<anchor synch="#TS-t02"/>but I never inhaled the smoke</u>
<u who="#bob">
  <anchor synch="#TS-t01"/>You used to smoke<anchor synch="#TS-t02"/>
</u>
```

As above, since the whole of Bob's utterance is to be aligned, the **start** and **end** attributes may be used as an alternative to the second pair of <anchor> elements:

```
<u start="#TS-t01" end="#TS-t02" who="#bob">You used to smoke</u>
```

An alternative approach is to mark the synchronization by pointing from the <timeline> to the text:

```
<timeline origin="#TS-T01">
  <when synch="#TS-nm1 #bob-u2" xml:id="TS-T01"/>
  <when synch="#TS-nm2 #bob-u2" xml:id="TS-T02"/>
</timeline>
<u who="#tom">I used to smoke
<anchor xml:id="TS-nm1"/>a lot more than this
<anchor xml:id="TS-nm2"/>but I never inhaled the smoke</u>
<u xml:id="bob-u2" who="#bob">You used to smoke</u>
```

To avoid deciding whether to point from the timeline to the text or vice versa, a <linkGrp> may be used:

```
<body>
  <timeline origin="#T001">
    <when xml:id="T001"/>
    <when xml:id="T002"/>
  </timeline>
  <u who="#tom">I used to smoke
  <anchor xml:id="NM01"/>a lot more than this
  <anchor xml:id="NM02"/>but I never inhaled the smoke</u>
  <u xml:id="bob-U2" who="#bob">You used to smoke</u>
  <linkGrp type="synchronize">
    <link targets="#T001 #NM01 #bob-U2"/>
    <link targets="#T002 #NM02 #bob-U2"/>
  </linkGrp>
</body>
```

Note that in each case, although Bob's utterance follows Tom's sequentially in the text, it is aligned temporally with its middle, without any need to disrupt the normal syntax of the text.

As a final example, consider the following exchange, first as it might be represented using a musical-score-like notation, in which points of synchronization are represented by vertical alignment of the text:

```
A : This is |my |turn
B : |Balderdash
C : |No, |it's mine
```

All three speakers are simultaneous at the words *my*, *Balderdash*, and *No*; speakers A and C are simultaneous at the words *turn* and *it's*. This could be encoded as follows, using pointers from the alignment map into the text:

```
<timeline origin="#TSp1">
  <when synch="#TSa1 #TSb1 #TSc1" xml:id="TSp1"/>
  <when synch="#TSa2 #TSc2" xml:id="TSp2"/>
</timeline>
<u who="#A">this is <anchor xml:id="TSa1"/> my <anchor xml:id="TSa2"/> turn</u>
<u who="#B" xml:id="TSb1">balderdash</u>
<u who="#C" xml:id="TSc1"> no <anchor xml:id="TSc2"/> it's mine</u>
```

8.4.3 Regularization of Word Forms

When speech is transcribed using ordinary orthographic notation, as is customary, some compromise must be made between the sounds produced and conventional orthography. Particularly when dealing with informal, dialectal, or other varieties of language, the transcriber will frequently have to decide whether a particular sound is to be treated as a distinct vocabulary item or not. For example, while in a given project *kinda* may not be worth distinguishing as a vocabulary item from *kind of*, *isn't* may clearly be worth distinguishing from *is not*; for some purposes, the regional variant *isnae* might also be worth distinguishing in the same way.

One rule of thumb might be to allow such variation only where a generally accepted orthographic form exists, for example, in published dictionaries of the language register being encoded; this has the disadvantage that such dictionaries may not exist. Another is to maintain a controlled (but extensible) set of normalized forms for all such words; this has the advantage of enforcing some degree of consistency among different transcribers. Occasionally, as for example when transcribing abbreviations or acronyms, it may be felt necessary to depart from conventional spelling to distinguish between cases where the abbreviation is spelled out letter by letter (e.g. *B B C* or *V A T*) and where it is pronounced as a single word (*VAT* or *RADA*). Similar considerations might apply to pronunciation of foreign words (e.g. *Monsewer* vs. *Monsieur*).

In general, use of punctuation, capitalization, etc., in spoken transcripts should be carefully controlled. It is important to distinguish the transcriber's intuition as to what the punctuation should be from the marking of prosodic features such as pausing, intonation, etc.

Whatever practice is adopted, it is essential that it be clearly and fully documented in the editorial declarations section of the header. It may also be found helpful to include normalized forms of non-conventional spellings within the text, using the elements for simple editorial changes described in section *3.4. Simple Editorial Changes* (see further section *8.4.5. Speech Management*).

8.4.4 Prosody

In the absence of conventional punctuation, the marking of prosodic features assumes paramount importance, since these structure and organize the spoken message. Indeed, such prosodic features as points of primary or secondary stress may be represented by specialized punctuation marks, or other characters such as those provided by the Unicode Spacing Modifier Letters block. Pauses have already been dealt with in section *8.3.2. Pausing*; while tone units (or intonational phrases) can be indicated by the segmentation tag discussed in section *8.4.1. Segmentation*. The <shift> element discussed in section *8.3.6. Shifts* may also be used to encode some prosodic features, for example where all that is required is the ability to record shifts in voice quality.

In a more detailed phonological transcript, it is common practice to include a number of conventional signs to mark prosodic features of the surrounding or (more usually) preceding speech. Such signs may be

used to record, for example, particular intonation patterns, truncation, vowel quality (long or short) etc. These signs may be preserved in a transcript either by using conventional punctuation or by marking their presence by <g> elements. Where a transcript includes many phonetic or phonemic aspects, it will generally be more convenient to use the appropriate Unicode characters (see further chapters *vi Languages and Character Sets* and *5. Representation of Non-standard Characters and Glyphs.* For representation of phonemic information, the use of the International Phonetic Alphabet, which can be represented in Unicode characters, is recommended.

In the following example, special characters have been defined as follows within the <encodingDesc> of the TEI header

```
<charDecl>
  <char xml:id="lf">
    <desc>low fall intonation</desc>
  </char>
  <char xml:id="lr">
    <desc>low rise intonation</desc>
  </char>
  <char xml:id="fr">
    <desc>fall rise intonation</desc>
  </char>
  <char xml:id="rf">
    <desc>rise fall intonation</desc>
  </char>
  <char xml:id="long">
    <desc>lengthened syllable</desc>
  </char>
  <char xml:id="short">
    <desc>shortened syllable</desc>
  </char>
</charDecl>
```

These declarations might additionally provide information about how the characters concerned should be rendered, their equivalent IPA form, etc. In the transcript itself references to them can then be included as follows:

```
<div n="Lod E-03" type="exchange">
  <note>C is with a friend</note>
  <u who="#cwn">
    <unclear>Excuse me<g ref="#lf"/>
    </unclear>
    <pause/> You dont have some
     aesthetic<g ref="#short"/>
    <pause/>
    <unclear>specially on early</unclear>
     aesthetics terminology <g ref="#lr"/>
  </u>
  <u who="#aj"> No<g ref="#lf"/>
    <pause/>No<g ref="#lf"/>
    <gap extent="2"/> I'm
     afraid<g ref="#lf"/>
  </u>
  <u trans="latching" who="#cwn"> No<g ref="#lr"/>
    <unclear>Well</unclear> thanks<g ref="#lr"/>
    <pause/> Oh<g ref="#short"/>
    <unclear>you couldnt<g ref="#short"/> can we</unclear> kind of<g ref="#long"/>
```

```
    <pause/>I mean ask you to order it for us<g  ref="#long"/>
    <g  ref="#fr"/>
  </u>
  <u  trans="latching"  who="#aj"> Yes<g  ref="#fr"/> if you know the title<g  ref="#lf"/> Yeah<g  ref="#lf"/>
  </u>
  <u  who="#cwn">
    <gap  extent="3"/>
    <gap  extent="4"/>
  </u>
  <u  who="#aj"> Yes thats fine. <unclear>just as soon as it comes in we'll send
        you a postcard<g  ref="#lf"/>
    </unclear>
  </u>
  <listPerson>
    <person  xml:id="cwn">
      <p>Customer WN</p>
    </person>
    <person  xml:id="aj">
      <p>Assistant K</p>
    </person>
  </listPerson>
</div>
```

Source: [79]

This example, which is taken from a corpus of bookshop service encounters, also demonstrates the use of the <unclear> and <gap> elements discussed in section *3.4. Simple Editorial Changes*. Where words are so unclear that only their extent can be recorded, the empty <gap> element may be used; where the encoder can identify the words but wishes to record a degree of uncertainty about their accuracy, the <unclear> element may be used. More flexible and detailed methods of indicating uncertainty are discussed in chapter *21. Certainty and Responsibility*.

For more detailed work, involving a detailed phonological transcript including representation of stress and pitch patterns, it is probably best to maintain the prosodic description in parallel with the conventional written transcript, rather than attempt to embed detailed prosodic information within it. The two parallel streams may be aligned with each other and with other streams, for example an acoustic encoding, using the general alignment mechanisms discussed in section *8.3.6. Shifts*.

8.4.5 Speech Management

Phenomena of speech management include disfluencies such as filled and unfilled pauses, interrupted or repeated words, corrections, and reformulations as well as interactional devices asking for or providing feedback. Depending on the importance attached to such features, transcribers may choose to adopt conventionalized representations for them (as discussed in section *8.4.3. Regularization of Word Forms* above), or to transcribe them using IPA or some other transcription system. To simplify analysis of the lexical features of a speech transcript, it may be felt useful to 'tidy away' many of these disfluencies. Where this policy has been adopted, these Guidelines recommend the use of the tags for simple editorial intervention discussed in section *3.4. Simple Editorial Changes*, to make explicit the extent of regularization or normalization performed by the transcriber.

For example, false starts, repetition, and truncated words might all be included within a transcript, but marked as editorially deleted, in the following way:

```
<u>
  <del  type="truncation">s</del>see
```

```
<del type="repetition">you you</del> you know
<del type="falseStart">it's</del> he's crazy
</u>
```

As previously noted, the <gap> element may be used to mark points within a transcript where words have been omitted, for example because they are inaudible, as in the following example in which 5 seconds of speech is drowned out by an external event:

```
<gap reason="passing truck" extent="5" unit="s"/>
```

The <unclear> element may be used to mark words which have been included although the transcriber is unsure of their accuracy:

```
<u>...and then <unclear reason="passing truck">marbled queen</unclear>
</u>
```

Where a transcriber is believed to have incorrectly identified a word, the elements <corr> or <sic> embedded within a <choice> element may be used to indicate both the original and a corrected form of it:

```
<choice>
  <corr>SCSI</corr>
  <sic>skuzzy</sic>
</choice>
```

These elements are further discussed in section *3.4.1. Apparent Errors*.

Finally phenomena such as code-switching, where a speaker switches from one language to another, may easily be represented in a transcript by using the <foreign> element provided by the core tagset:

```
<u who="#P1">I proposed that <foreign xml:lang="de"> wir können
  <pause dur="PT1S"/> vielleicht </foreign> go to warsaw
and <emph>vienna</emph>
</u>
```

8.4.6 Analytic Coding

The recommendations made here only concern the establishment of a basic text. Where a more sophisticated analysis is needed, more sophisticated methods of markup will also be appropriate, for example, using stand-off markup to indicate multiple segmentation of the stream of discourse, or complex alignment of several segments within it. Where additional annotations (sometimes called 'codes' or 'tags') are used to represent such features as linguistic word class (noun, verb, etc.), type of speech act (imperative, concessive, etc.), or information status (theme/rheme, given/new, active/semi-active/new), etc., a selection from the general purpose analytic tools discussed in chapters *16. Linking, Segmentation, and Alignment*, *17. Simple Analytic Mechanisms*, and *18. Feature Structures*, may be used to advantage.

8.5 Module for Transcribed Speech

The module described in this chapter makes available the following components:

Module spoken: Transcribed Speech

- Elements defined: broadcast equipment incident kinesic pause recording recordingStmt scriptStmt shift u vocal writing
- Classes defined: att.duration model.divPart.spoken model.global.spoken model.recordingPart

The selection and combination of modules to form a TEI schema is described in *1.2. Defining a TEI Schema*.

Chapter 9

Dictionaries

This chapter defines a module for encoding human-oriented monolingual and multilingual dictionaries, glossaries, and similar documents. The elements described here may also be useful in the encoding of computational lexica and similar resources intended for use by language-processing software; they may also be used to provide a rich encoding for wordlists, lexica, glossaries, etc. included within other documents. Dictionaries are most familiar in their printed form; however, increasing numbers of dictionaries exist also in electronic forms which are independent of any particular printed form, but from which various displays can be produced.

Both typographically and structurally, print dictionaries are extremely complex. In addition, dictionaries are of interest to many communities with different and sometimes conflicting goals. As a result, many general problems of text encoding are particularly pronounced here, and more compromises and alternatives within the encoding scheme may be required in future.[1] Two problems are particularly prominent.

First, because the structure of dictionary entries varies widely both among and within dictionaries, the simplest way for an encoding scheme to accommodate the entire range of structures actually encountered is to allow virtually any element to appear virtually anywhere in a dictionary entry. It is clear, however, that strong and consistent structural principles do govern the vast majority of conventional dictionaries, as well as many or most entries even in more 'exotic' dictionaries; encoding guidelines should include these structural principles. We therefore define two distinct elements for dictionary entries, one (<entry>) which captures the regularities of many conventional dictionary entries, and a second (<entryFree>) which uses the same elements, but allows them to combine much more freely. It is however recommended that <entry> be used in preference to <entryFree> wherever possible. These elements and their contents are described in sections *9.2. The Structure of Dictionary Entries*, *9.6. Unstructured Entries*, and *9.4. Headword and Pronunciation References*.

Second, since so much of the information in printed dictionaries is implicit or highly compressed, their encoding requires clear thought about whether it is to capture the precise typographic form of the source text or the underlying structure of the information it presents. Since both of these views of the dictionary may be of interest, it proves necessary to develop methods of recording both, and of recording the interrelationship between them as well. Users interested mainly in the printed format of the dictionary will require an encoding to be faithful to an original printed version. However, other users will be interested primarily in capturing the lexical information in a dictionary in a form suitable for further processing, which may demand the expansion or rearrangement of the information contained in the printed form. Further, some users wish to encode *both* of these views of the data, and retain the links between related elements of the two encodings. Problems of recording these two different views of dictionary data are discussed in section *9.5. Typographic and Lexical Information in Dictionary Data*, together with mechanisms for retaining both views when this is desired.

[1] We refer the reader to previous and current discussions of a common format for encoding dictionaries. For example, Amsler and Tompa (1988); Calzolari et al. (1990);Fought and Van Ess-Dykema; Ide and Veronis (1995); Ide et al. (1993); Ide et al. (1992); DANLEX Group (1987); and Tutin and Veronis (1998); Ide et al. (2000).

To deal with this complexity, and in particular to account for the wide variety of linguistic context within which a dictionary may be designed, it can be necessary to customize or change the schema by providing more restriction or possibly alternate content models for the elements defined in this chapter. Section *9.3.2. Grammatical Information* illustrates this with the provision of a closed set of values for grammatical descriptors.

This chapter contains a large number of examples taken from existing print dictionaries; in each case, the original source is identified. In presenting such examples, we have tried to retain the original typographic appearance of the example as well as presenting a suggested encoding for it. Where this has not been possible (for example in the display of pronounciation) we have adopted the transliteration found in the electronic edition of the *Oxford Advanced Learner's Dictionary*. Also, the middle dot in quoted entries is rendered with a full stop, while within the sample transcriptions hyphenation and syllabification points are indicated by a vertical bar |, regardless of their appearance in the source text.

9.1 Dictionary Body and Overall Structure

Overall, dictionaries have the same structure of front matter, body, and back matter familiar from other texts. In addition, this modules defines <entry>, <entryFree>, and <superEntry> as component-level elements which can occur directly within a text division or the text body.

The following tags can therefore be used to mark the gross structure of a printed dictionary; the dictionary-specific tags are discussed further in the following section.

<text> contains a single text of any kind, whether unitary or composite, for example a poem or drama, a collection of essays, a novel, a dictionary, or a corpus sample.

<front> (front matter) contains any prefatory matter (headers, title page, prefaces, dedications, etc.) found at the start of a document, before the main body.

<body> (text body) contains the whole body of a single unitary text, excluding any front or back matter.

<back> (back matter) contains any appendixes, etc. following the main part of a text.

** ** (text division) contains a subdivision of the front, body, or back of a text.

<entry> contains a reasonably well-structured dictionary entry.

<entryFree> (unstructured entry) contains a dictionary entry which does not necessarily conform to the constraints imposed by the <entry> element.

<superEntry> groups successive entries for a set of homographs.

As members of the class att.entryLike, <entry> and <entryFree> share the following attributes:

att.entryLike groups the different styles of dictionary entries.

@**type** indicates type of entry, in dictionaries with multiple types.

@**sortKey** contains a (sortable) character sequence reflecting the entry's alphabetical position in the printed dictionary.

The front and back matter of a dictionary may well contain specialized material such as lists of common and proper nouns, grammatical tables, gazetteers, a 'guide to the use of the dictionary', etc. These should be tagged using elements defined elsewhere in these Guidelines, chiefly in the core module (chapter *3. Elements Available in All TEI Documents*) together with the specialized dictionary elements defined in this chapter.

The <body> element consists of a set of entries, optionally grouped into one or several elements. These text divisions might correspond, for example, sections for different letters of the alphabet, or to sections for different languages in bilingual dictionaries, as in the following example:

```
<body>
  <div>
    <head>English-French</head>
    <entry>
```

```
<!-- ... -->
   </entry>
   <entry>
<!-- ... -->
   </entry>
   <entry>
<!-- ... -->
   </entry>
  </div>
  <div>
   <head>French-English</head>
   <entry>
<!-- ... -->
   </entry>
   <entry>
<!-- ... -->
   </entry>
   <entry>
<!-- ... -->
   </entry>
  </div>
</body>
```

In a print dictionary, the entries are typically typographically distinct entities, each headed by some morphological form of the lexical item described (the headword), and sorted in alphabetical order or (especially for non-alphabetic scripts) in some other conventional sequence. Dictionary entries should be encoded as distinct successive items, each marked as an <entry> or <entryFree> element. The **type** attribute may be used to distinguish different types of entries, for example main entries, related entries, run-on entries, or entries for cross-references, etc.

Some dictionaries provide distinct entries for homographs, on the basis of etymology, part-of-speech, or both, and typically provide a numeric superscript on the headword identifying the homograph number. In these cases each homograph should be encoded as a separate entry; the <superEntry> element may optionally be used to group such successive homograph entries. In addition to a series of <entry> elements, the <superEntry> may contain a preliminary <form> group (see section *9.3.1. Information on Written and Spoken Forms*) when information about hyphenation, pronunciation, etc., is given only once for two or more homograph entries. If the homograph number is to be recorded, the global attribute **n** may be used for this purpose. In some dictionaries, homographs are treated in distinct parts of the same entry; in these cases, they may be separated by use of the <hom> element, for which see section *9.2.1. Hierarchical Levels*.

A sort key, given in the **key** attribute, is often required for superentries and entries, especially in cases where the order of entries does not follow the local character-set collating sequence (as, for example, when an entry for '3D' appears at the place where 'three-D' would appear).

A dictionary with no internal divisions might thus have a structure like the following; a <superEntry> is shown grouping two homograph entries.

```
<body>
  <entry>
<!-- ... -->
  </entry>
  <entry>
<!-- ... -->
  </entry>
```

```
  <superEntry>
    <entry  type="hom"  n="1"/>
    <entry  type="hom"  n="2"/>
  </superEntry>
</body>
```

9.2 The Structure of Dictionary Entries

A simple dictionary entry may contain information about the form of the word treated, its grammatical characterization, its definition, synonyms, or translation equivalents, its etymology, cross-references to other entries, usage information, and examples. These we refer to as the constituent parts or constituents of the entry; some dictionary constituents possess no internal structure, while others are most naturally viewed as groups of smaller elements, which may be marked in their own right. In some styles of markup, tags will be applied only to the low-level items, leaving the constituent groups which contain them untagged. We distinguish the class of top-level constituents of dictionary entries, which can occur directly within entries, from the class of phrase-level constituents, which can normally occur only within top-level constituents. The top-level constituents of dictionary entries are described in section *9.2.2. Groups and Constituents*, and documented more fully, together with their phrase-level sub-constituents, in section *9.3. Top-level Constituents of Entries*.

In addition, however, dictionary entries often have a complex hierarchical structure. For example, an entry may consist of two or more sub-parts, each corresponding to information for a different part-of-speech homograph of the headword. The entry (or part-of-speech homographs, if the entry is split this way) may also consist of senses, each of which may in turn be composed of two or more sub-senses, etc. Each sub-part, homograph entry, sense, or sub-sense we call a level; at any level in an entry, any or all of the constituent parts of dictionary entries may appear. The hierarchical levels of dictionary entries are documented in section *9.2.1. Hierarchical Levels*.

9.2.1 Hierarchical Levels

The outermost structural level of an entry is marked with the elements <entry> or <entryFree>. The <hom> element marks the subdivision of entries into homographs differing in their part-of-speech. The <sense> element marks the subdivision of entries and part-of-speech homographs into senses; this element nests recursively in order to provide for a hierarchy of sub-senses of any depth. All of these levels may each contain any of the constituent parts of an entry. A special case of hierarchical structure is represented by the <re> (related entry) element, which is discussed in section *9.3.6. Related Entries*. Finally, the element <dictScrap> may be used at any point in the hierarchy to delimit parts of the dictionary entry which are structurally anomalous, as further discussed in section *9.6. Unstructured Entries*.

<entry> contains a reasonably well-structured dictionary entry.

<entryFree> (unstructured entry) contains a dictionary entry which does not necessarily conform to the constraints imposed by the <entry> element.

<hom> (homograph) groups information relating to one homograph within an entry.

<sense> groups together all information relating to one word sense in a dictionary entry, for example definitions, examples, and translation equivalents.

@**level** gives the nesting depth of this sense.

<dictScrap> (dictionary scrap) encloses a part of a dictionary entry in which other phrase-level dictionary elements are freely combined.

For example, an entry with two senses will have the following structure:

```
<entry>
  <sense  n="1"/>
  <sense  n="2"/>
</entry>
```

An entry with two homographs, the first with two senses and the second with three (one of which has two sub-senses), may have a structure like this:

```
<entry>
  <hom  n="1">
    <sense  n="1">
<!-- ... -->
    </sense>
    <sense  n="2">
<!-- ... -->
    </sense>
  </hom>
  <hom  n="2">
    <sense  n="1">
      <sense  n="a">
<!-- ... -->
      </sense>
      <sense  n="b">
<!-- ... -->
      </sense>
    </sense>
    <sense  n="2">
<!-- ... -->
    </sense>
    <sense  n="3">
<!-- ... -->
    </sense>
  </hom>
</entry>
```

In some dictionaries, homographs have separate entries; in such a case, as noted in section *9.1. Dictionary Body and Overall Structure*, the two homographs may be treated as entries, optionally grouped in a <superEntry>:

```
<superEntry>
  <entry  n="1"  type="hom">
    <sense  n="1">
<!-- ... -->
    </sense>
    <sense  n="2">
<!-- ... -->
    </sense>
  </entry>
  <entry  n="2"  type="hom">
    <sense  n="1">
      <sense  n="a">
<!-- ... -->
      </sense>
      <sense  n="b">
<!-- ... -->
```

```
     </sense>
    </sense>
    <sense n="2">
<!-- ... -->
    </sense>
    <sense n="3">
<!-- ... -->
    </sense>
  </entry>
</superEntry>
```

The hierarchic structure of a dictionary entry is enforced by the structures defined in this module. The content model for <entry> specifies that entries do not nest, that homographs nest within entries, and that senses nest within entries, homographs, or senses, and may be nested to any depth to reflect the embedding of sub-senses. Any of the top-level constituents (<def>, <usg>, <form>, etc.) can appear at any level (i.e., within entries, homographs, or senses).

9.2.2 Groups and Constituents

As noted above, dictionary entries, and subordinate levels within dictionary entries, may comprise several constituent parts, each providing a different type of information about the word treated. The top-level constituents of dictionary entries are:

- information about the form of the word treated (orthography, pronunciation, hyphenation, etc.)
- grammatical information (part of speech, grammatical sub-categorization, etc.)
- definitions or translations into another language
- etymology
- examples
- usage information
- cross-references to other entries
- notes
- entries (often of reduced form) for related words, typically called related entries

Any of the hierarchical levels (<entry>, <entryFree>, <hom>, and <sense>) may contain any of these top-level constituents, since information about word form, particular grammatical information, special pronunciation, usage information, etc., may apply to an entire entry, or to only one homograph, or only to a particular sense. The examples below illustrate this point.

The following elements are used to encode these top-level constituents:

<form> (form information group) groups all the information on the written and spoken forms of one headword.

<gramGrp> (grammatical information group) groups morpho-syntactic information about a lexical item, e.g. <pos>, <gen>, <number>, <case>, or <iType> (inflectional class).

<def> (definition) contains definition text in a dictionary entry.

<cit> (cited quotation) contains a quotation from some other document, together with a bibliographic reference to its source. In a dictionary it may contain an example text with at least one occurrence of the word form, used in the sense being described, or a translation of the headword, or an example.

<usg> (usage) contains usage information in a dictionary entry.

<xr> (cross-reference phrase) contains a phrase, sentence, or icon referring the reader to some other location in this or another text.

<etym> (etymology) encloses the etymological information in a dictionary entry.

<re> (related entry) contains a dictionary entry for a lexical item related to the headword, such as a compound phrase or derived form, embedded inside a larger entry.

<note> contains a note or annotation.

In a simple entry with no internal hierarchy, all top-level constituents appear at the <entry> level.

> **com.peti.tor**/k@m"petit@(r)/ n person who competes. OALD

```
<entry>
  <form>
    <orth>competitor</orth>
    <hyph>com|peti|tor</hyph>
    <pron>k@m"petit@(r)</pron>
  </form>
  <gramGrp>
    <pos>n</pos>
  </gramGrp>
  <def>person who competes.</def>
</entry>
```

For the elements which appear within the <form> and <gramGrp> elements of this and other examples, see below, section *9.3.1. Information on Written and Spoken Forms*, and section*9.3.2. Grammatical Information*.

Any top-level constituent can appear at any level when the hierarchical structure of the entry is more complex. The most obvious examples are <def> and <cit>, which appear at the <sense> level when several senses or translations exist:

> **disproof**(dIs"pru:f) n. 1. facts that disprove something. 2. the act of disproving. CED

```
<entry>
  <form>
    <orth>disproof</orth>
    <pron>dIs"pru:f</pron>
  </form>
  <gramGrp>
    <pos>n</pos>
  </gramGrp>
  <sense  n="1">
    <def>facts that disprove something.</def>
  </sense>
  <sense  n="2">
    <def>the act of disproving.</def>
  </sense>
</entry>
```

In the following example, <gramGrp> is used to distinguish two homographs:

> **bray**/breI/ n cry of an ass; sound of a trumpet. · vt [VP2A] make a cry or sound of this kind. OALD

```
<entry>
  <form>
    <orth>bray</orth>
    <pron>breI</pron>
  </form>
  <hom>
    <gramGrp>
      <pos>n</pos>
    </gramGrp>
    <def>cry of an ass; sound of a trumpet.</def>
  </hom>
  <hom>
    <gramGrp>
      <pos>vt</pos>
      <subc>VP2A</subc>
    </gramGrp>
    <def>make a cry or sound of this kind.</def>
  </hom>
</entry>
```

Information of the same kind can appear at different levels within the same entry; here, grammatical information occurs both at entry and homograph level.

> **ca.reen**/k@"ri:n/ vt,vi 1 [VP6A] turn (a ship) on one side for cleaning, repairing, etc. 2 [VP6A, 2A] (cause to) tilt, lean over to one side. OALD

```
<entry>
  <form>
    <orth>careen</orth>
    <hyph>ca|reen</hyph>
    <pron>k@"ri:n</pron>
  </form>
  <gramGrp>
    <pos>vt</pos>
    <pos>vi</pos>
  </gramGrp>
  <sense  n="1">
    <gramGrp>
      <subc>VP6A</subc>
    </gramGrp>
    <def>turn (a ship) on one side for cleaning, repairing, etc.</def>
  </sense>
  <sense  n="2">
    <gramGrp>
      <subc>VP6A</subc>
      <subc>VP2A</subc>
    </gramGrp>
    <def>(cause to) tilt, lean over to one side.</def>
  </sense>
</entry>
```

Alone among the constituent groups, <form> can appear at the <superEntry> level as well as at the <entry>, <hom>, and <sense> levels:

a.ban.don 1/@"band@n/ v [T1] 1 to leave completely and for ever; desert: The sailors abandoned the burning ship. 2 …**abandon** 2 n [U] the state when one's feelings and actions are uncontrolled; freedom from control...LDOCE

```
<superEntry>
  <form>
    <orth>abandon</orth>
    <hyph>a|ban|don</hyph>
    <pron>@"band@n</pron>
  </form>
  <entry n="1">
    <gramGrp>
      <pos>v</pos>
      <subc>T1</subc>
    </gramGrp>
    <sense n="1">
      <def>to leave completely and for ever … </def>
    </sense>
    <sense n="2"/>
  </entry>
  <entry n="2">
    <gramGrp>
      <pos>n</pos>
      <subc>U</subc>
    </gramGrp>
    <def>the state when one's feelings and actions are uncontrolled; freedom
        from control…</def>
  </entry>
</superEntry>
```

9.3 Top-level Constituents of Entries

This section describes the top-level constituents of dictionary entries, together with the phrase-level constituents peculiar to each.

- the <form> element, which groups orthographic information and pronunciations, is described in section *9.3.1. Information on Written and Spoken Forms*
- the <gramGrp> element, which groups elements for the grammatical characterization of the headword, is described in section *9.3.2. Grammatical Information*
- the <def> element, which describes the meaning of the headword, is described in section *9.3.3. Sense Information*
- the <etym> element and its special phrase-level elements are documented in section *9.3.4. Etymological Information*
- the <cit> element and its specific applications are described in section *9.3.3. Sense Information* and section *9.3.5. Other Information*
- the <usg>, <lbl>, <xr>, and <note> elements are described in section *9.3.5. Other Information*
- the <re> element, which marks nested entries for related words, is described in section *9.3.6. Related Entries*

9.3.1 Information on Written and Spoken Forms

Dictionary entries most often begin with information about the form of the word to which the entry applies. Typically, the orthographic form of the word, sometimes marked for syllabification or hyphenation, is the first

item in an entry. Other information about the word, including variant or alternate forms, inflected forms, pronunciation, etc., is also often given.

The following elements should be used to encode this information: the <form> element groups one or more occurrences of any of them; it can also be recursively nested to reflect more complex sub-grouping of information about word form(s), as shown in the examples.

<form> (form information group) groups all the information on the written and spoken forms of one headword.

@**type** classifies form as simple, compound, etc.

<orth> (orthographic form) gives the orthographic form of a dictionary headword.

@**type** gives the type of spelling.

@**extent** gives the extent of the orthographic information provided.

<pron> (pronunciation) contains the pronunciation(s) of the word.

@**extent** indicates whether the pronunciation is for whole word or part.

<hyph> (hyphenation) contains a hyphenated form of a dictionary headword, or hyphenation information in some other form.

<syll> (syllabification) contains the syllabification of the headword.

<stress> contains the stress pattern for a dictionary headword, if given separately.

<lbl> (label) contains a label for a form, example, translation, or other piece of information, e.g. abbreviation for, contraction of, literally, approximately, synonyms:, etc.

In addition to those listed above, the following elements, which encode morphological details of the form, may also occur within <form> elements:

<gram> (grammatical information) within an entry in a dictionary or a terminological data file, contains grammatical information relating to a term, word, or form.

@**type** classifies the grammatical information given according to some convenient typology — in the case of terminological information, preferably the dictionary of data element types specified in ISO WD 12 620.

<gen> (gender) identifies the morphological gender of a lexical item, as given in the dictionary.

<number> indicates grammatical number associated with a form, as given in a dictionary.

<case> contains grammatical case information given by a dictionary for a given form.

<per> (person) contains an indication of the grammatical person (1st, 2nd, 3rd, etc.) associated with a given inflected form in a dictionary.

<tns> (tense) indicates the grammatical tense associated with a given inflected form in a dictionary.

<mood> contains information about the grammatical mood of verbs (e.g. indicative, subjunctive, imperative).

<iType> (inflectional class) indicates the inflectional class associated with a lexical item.

@**type** indicates the type of indicator used to specify the inflection class, when it is necessary to distinguish between the usual abbreviated indications (e.g. *inv*) and other kinds of indicators, such as special codes referring to conjugation patterns, etc.

Of these, the <gram> element is most general, and all of the others are synonymous with a <gram> element with appropriate values (**gen, number, case**, etc.) for the **type** attribute.

Different dictionaries use different means to mark hyphenation, syllabification, and stress, and they often use some unusual glyphs (e.g., the 'middle dot' for hyphenation). All of these glyphs are in the Unicode character set, as discussed in *v.6.1 Character References*. When transcribing representations of pronunciation the International Phonetic Alphabet should be used. It may be convenient (as has been done in the text of this

chapter) to use a simple transliteration scheme for this; such a scheme should however be properly documented in the header.

In the simplest case, nothing is given but the orthography:

```
<form>
  <orth>doom-laden</orth>
</form>
```

Often, however, pronunciation is given.

soucoupe [sukup] … DNT

```
<form>
  <orth>soucoupe</orth>
  <pron>sukup</pron>
</form>
```

For a variety of reasons including ease of processing, it may be desired to split into separate elements information which is collapsed into a single element in the source text; orthography and hyphenation may for example be transcribed as separate elements, although given together in the source text. For a discussion of the issues involved, and of methods for retaining both the presentation form and the interpreted form, see section *9.5. Typographic and Lexical Information in Dictionary Data.*

This example splits orthography and hyphenation, and adds syllabification because it differs from hyphenation:

ar.ea … W7

```
<form>
  <orth>area</orth>
  <hyph>ar|ea</hyph>
  <syll>ar|e|a</syll>
</form>
```

Multiple orthographic forms may be given, e.g. to illustrate a word's inflectional pattern:

brag … vb. brags, bragging, bragged … CED

```
<form>
  <orth>brag</orth>
</form>
<gramGrp>
  <pos>vb</pos>
</gramGrp>
<form type="infl">
  <orth>brags</orth>
  <orth>bragging</orth>
  <orth>bragged</orth>
</form>
```

Or the inflectional pattern may be indicated by reference to a table of paradigms, as here:

horrifier`[ORifje]` (7) vt … [C/R]

```
<form>
  <orth>horrifier</orth>
  <pron>ORifje</pron>
  <iType  type="vbtable">7</iType>
</form>
```

Explanatory labels may be attached to alternate forms:

MTBF *abbrev. for* mean time between failures. CED

```
<entry>
  <form  type="abbrev">
    <orth>MTBF</orth>
  </form>
  <form  type="full">
    <lbl>abbrev. for</lbl>
    <orth>mean time between failures</orth>
  </form>
</entry>
```

When multiple orthographic forms are given, a pronunciation may be associated with all of them, as here:

biryani or **biriani**`(%bIrI"A:nI)` … CED

```
<form>
  <orth>biryani</orth>
  <orth>biriani</orth>
  <pron>%bIrI"A:nI</pron>
</form>
```

In other cases, different pronunciations are provided for different orthographic forms; here, the <form> element is repeated to associate the first orthographic form explicitly with the first pronunciation, and the second orthographic form with the second pronunciation:

mackle`("mak^@l)` or macule `("makju:l)` … CED

```
<form>
  <orth>mackle</orth>
  <pron>"makəl</pron>
</form>
<form>
  <orth>macule</orth>
  <pron>"makju:l</pron>
</form>
```

Recursive nesting of the <form> element can preserve relations among elements that are implicit in the text. For example, in the CED entry for 'hospitaller', it is clear that 'U.S.' is associated only with 'hospitaler', but that the pronunciation applies to both forms. The following encoding preserves these relations:

hospitaller or U.S. hospitaler ("hQspIt@l@) … CED

```
<form>
  <orth>hospitaller</orth>
  <form>
    <usg type="geo">U.S.</usg>
    <orth>hospitaler</orth>
  </form>
  <pron>"hQspIt@l@</pron>
</form>
```

9.3.2 Grammatical Information

The <gramGrp> element groups grammatical information, such as part of speech, subcategorization information (e.g., syntactic patterns for verbs, count/mass distinctions for nouns), etc. It can contain any of the following elements:

<pos> (part of speech) indicates the part of speech assigned to a dictionary headword such as noun, verb, or adjective.

<subc> (subcategorization) contains subcategorization information (transitive/intransitive, countable/non-countable, etc.)

<colloc> (collocate) contains a collocate of the headword.

In addition, <gramGrp> can contain any of the morphological elements defined in section *9.3.1. Information on Written and Spoken Forms* for <form>. Elements conveying morphological information bear different interpretations within <gramGrp> and <form> groups, the difference being that in the <form> group, the morphological information specified pertains to the specific alternate form in question, while within <gramGrp> it applies to the headword form. For example, in the entry '**pinna** ('pIn@) n., pl. -nae (-ni:) or -nas'CED, the word defined can be either singular or plural; the 'pl.' specification applies only to the inflected forms provided. Compare this with 'pants (paents) pl. n.', where 'pl.' applies to the headword itself.

As noted above in section *9.3.1. Information on Written and Spoken Forms*, the elements for morphological information are simply shorthand for the general purpose <gram> element. Consider this entry for the French word *médire*:

médire v.t. ind. (de) … PLC

This entry can be tagged using specialized grammatical elements:

```
<form>
  <orth>médire</orth>
</form>
<gramGrp>
  <pos>v</pos>
  <subc>t ind</subc>
  <colloc type="prep">de</colloc>
</gramGrp>
```

Or using the <gram> element:

```
<form>
  <orth>médire</orth>
```

```
</form>
<gramGrp>
  <gram  type="pos">v</gram>
  <gram  type="subc">t ind</gram>
  <gram  type="collocPrep">de</gram>
</gramGrp>
```

Like <form>, <gramGrp> can be repeated, recursively nested, or used at the <sense> level to show relations among elements.

isotope adj. et n. m. … DNT

```
<form>
  <orth>isotope</orth>
</form>
<gramGrp>
  <pos>adj</pos>
</gramGrp>
<gramGrp>
  <pos>n</pos>
  <gen>m</gen>
</gramGrp>
```

wits (wIts) pl. n. 1. (sometimes sing.) the ability to reason and act, esp. quickly … CED

```
<entry>
  <form>
    <orth>wits</orth>
    <pron>wIts</pron>
  </form>
  <gramGrp>
    <number>pl</number>
    <pos>n</pos>
  </gramGrp>
  <sense  n="1">
    <gramGrp>
      <number>sometimes sing.</number>
    </gramGrp>
    <def>the ability to reason and act, esp.  quickly …</def>
  </sense>
</entry>
```

9.3.3 Sense Information

Dictionaries may describe the meanings of words in a wide variety of different ways — by means of synonyms, paraphrases, translations into other languages, formal definitions in various highly stylized forms, etc. No attempt is made here to distinguish all the different forms which sense information may take; all of them may be tagged using the <def> element described in section *9.3.3.1. Definitions*.

As a special case it is frequently desirable to distinguish the provision of translation equivalents in other languages from other forms of sense information; the use of **<cit type="translation">** (which groups a translation equivalent with related information such as its grammatical description) for this purpose is described in section *9.3.3.2. Translation Equivalents*.

9.3.3.1 Definitions

Dictionary definitions are those pieces of prose in a dictionary entry that describe the meaning of some lexical item. Most often, definitions describe the headword of the entry; in some cases, they describe translated texts, examples, etc.; see **<cit type="translation">**, section *9.3.3.2. Translation Equivalents*, and **<cit type="example">**, section *9.3.5.1. Examples*. The <def> element directly contains the text of the definition; unlike <form> and <gramGrp>, it does not serve solely to group a set of smaller elements. The close analysis of definition text, such as the tagging of hypernyms, typical objects, etc., is not covered by these Guidelines.

Definitions may occur directly within an entry; when multiple definitions are given, they are typically identified as belonging to distinct senses, as here:

> **demigod** (…) n. 1.a. a being who is part mortal, part god. b. a lesser deity. 2. a godlike person. CP

```
<entry>
  <form>
    <orth>demigod</orth>
    <pron> … </pron>
  </form>
  <gramGrp>
    <pos>n</pos>
  </gramGrp>
  <sense  n="1">
    <sense  n="a">
      <def>a being who is part mortal, part god.</def>
    </sense>
    <sense  n="b">
      <def>a lesser deity.</def>
    </sense>
  </sense>
  <sense  n="2">
    <def>a godlike person.</def>
  </sense>
</entry>
```

In multilingual dictionaries, it is sometimes possible to distinguish translation equivalents from definitions proper; here a <def> element is distinguished from the translation information within which it appears.

> **rémoulade**[Remulad] nf remoulade, rémoulade (*dressing containing mustard and herbs*). CR

```
<entry>
  <form>
    <orth>rémoulade</orth>
    <pron>Remulad</pron>
  </form>
  <gramGrp>
    <pos>n</pos>
    <gen>f</gen>
  </gramGrp>
  <cit  type="translation"  xml:lang="en">
    <quote>remoulade</quote>
    <quote>rémoulade</quote>
    <def>dressing containing mustard and herbs</def>
  </cit>
</entry>
```

9.3.3.2 Translation Equivalents

Multilingual dictionaries contain information about translations of a given word in some source language for one or more target languages. Minimally, the dictionary provides the corresponding translation in the target language; other material, such as morphological information (gender, case), various kinds of usage restrictions, etc., may also be given. If translation equivalents are to be distinguished from other kinds of sense information, they may be encoded using **<cit type="translation">**. The global **xml:lang** attribute should be used to specify the target language.

As in monolingual dictionaries, the <sense> element is used in multilingual dictionaries to group information (forms, grammatical information, usage, translation(s), etc.) about a given sense of a word where necessary. Information about the individual translation equivalents within a sense is grouped using **<cit type="translation">**. This information may include the translation text (tagged <q> or <quote>), morphological information (<gen>, <case>, etc.), usage notes (<usg>), translation labels (<lbl>), and definitions (<def>).When bibliographic data is provided, the <quote> element should be used.

<cit> (cited quotation) contains a quotation from some other document, together with a bibliographic reference to its source. In a dictionary it may contain an example text with at least one occurrence of the word form, used in the sense being described, or a translation of the headword, or an example.

<lbl> (label) contains a label for a form, example, translation, or other piece of information, e.g. abbreviation for, contraction of, literally, approximately, synonyms:, etc.

Note how in the following example, different translation equivalents are grouped into the same or different senses, following the punctuation of the source and the usage labels:

> **dresser** … (a) (Theat) habilleur m, -euse f; (Comm: window ~) étalagiste mf. she's a stylish ~ elle s'habille avec chic; V hair. (b) (tool) (for wood) raboteuse f; (for stone) rabotin m. CR

```
<entry n="1">
  <form>
    <orth>dresser</orth>
  </form>
  <sense n="a">
    <sense>
      <usg type="dom">Theat</usg>
      <cit type="translation" xml:lang="fr">
        <quote>habilleur</quote>
        <gen>m</gen>
      </cit>
      <cit type="translation" xml:lang="fr">
        <quote>-euse</quote>
        <gen>f</gen>
      </cit>
    </sense>
    <sense>
      <usg type="dom">Comm</usg>
      <form type="compound">
        <orth>window <oRef/>
        </orth>
      </form>
      <cit type="translation" xml:lang="fr">
        <quote>étalagiste</quote>
        <gen>mf</gen>
      </cit>
```

```
    </sense>
    <cit type="example">
      <quote>she's a stylish <oRef/>
      </quote>
      <cit type="translation" xml:lang="fr">
        <quote>elle s'habille avec chic</quote>
      </cit>
    </cit>
    <xr type="see">V. <ref target="#hair">hair</ref>
    </xr>
  </sense>
  <sense n="b">
    <usg type="category">tool</usg>
    <sense>
      <usg type="hint">for wood</usg>
      <cit type="translation" xml:lang="fr">
        <quote>raboteuse</quote>
        <gen>f</gen>
      </cit>
    </sense>
    <sense>
      <usg type="hint">for stone</usg>
      <cit type="translation" xml:lang="fr">
        <quote>rabotin</quote>
        <gen>m</gen>
      </cit>
    </sense>
  </sense>
</entry>
<!-- ... -->
<entry xml:id="hair">
<!-- ... -->
</entry>
```

In the following example, a distinction is made between the translation equivalent ('OAS') and a descriptive phrase providing further information for the user of the dictionary.

> **O.A.S.** ... nf (abrév de **Organisation de l'Armée secrète**) OAS (*illegal military organization supporting French rule of Algeria*). CR

```
<entry>
  <cit type="translation" xml:lang="en">
    <quote>OAS</quote>
    <def>illegal military organization supporting French rule of
        Algeria</def>
  </cit>
</entry>
```

Note that **<cit type="translation">** may also be used in monolingual dictionaries when a translation is given for a foreign word:

> **havdalah** or **havdoloh** Hebrew. (Hebrew `hAvdA"lA;` Yiddish `hAv"dOl@`) n. Judaism. the ceremony marking the end of the sabbath or of a festival, including the blessings over wine, candles and spices. [literally: separation] CED

```
<entry  type="foreign">
  <form>
    <orth>havdalah</orth>
    <orth>havdoloh</orth>
  </form>
  <usg  type="dom">Judaism</usg>
  <def>the ceremony marking the end of the sabbath or of a festival,
     including the blessings over wine, candles and spices.</def>
  <cit  type="translation"  xml:lang="en">
    <note>literally</note>
    <quote>separation</quote>
  </cit>
</entry>
```

9.3.4 Etymological Information

The element <etym> marks a block of etymological information. Etymologies may contain highly structured lists of words in an order indicating their descent from each other, but often also include related words and forms outside the direct line of descent, for comparison. Not infrequently, etymologies include commentary of various sorts, and can grow into short (or long!) essays with prose-like structure. This variation in structure makes it impracticable to define tags which capture the entire intellectual structure of the etymology or record the precise interrelation of all the words mentioned. It is, however, feasible to mark some of the more obvious phrase-level elements frequently found in etymologies, using tags defined in the core module or elsewhere in this chapter. Of particular relevance for the markup of etymologies are:

<etym> (etymology) encloses the etymological information in a dictionary entry.

<lang> (language name) name of a language mentioned in etymological or other linguistic discussion.

<date> contains a date in any format.

<mentioned> marks words or phrases mentioned, not used.

<gloss> identifies a phrase or word used to provide a gloss or definition for some other word or phrase.

<pron> (pronunciation) contains the pronunciation(s) of the word.

<usg> (usage) contains usage information in a dictionary entry.

<lbl> (label) contains a label for a form, example, translation, or other piece of information, e.g. abbreviation for, contraction of, literally, approximately, synonyms:, etc.

As in other prose, individual word forms mentioned in an etymological description are tagged with <mentioned> elements. Pronunciations, usage labels, and glosses can be tagged using the <pron>, <usg>, and <gloss> elements defined elsewhere in these Guidelines. In addition, the <lang> element may be used to identify a particular language name where it appears, in addition to using the **xml:lang** attribute of the <mentioned> element.

Examples:

> **abismo** m. (del gr. a priv. y byssos, fondo). Sima, gran profundidad. …

```
<entry>
  <form>
    <orth>abismo</orth>
  </form>
  <etym>del <lang>gr.</lang>
    <mentioned>a</mentioned> priv.  y <mentioned>byssos</mentioned>,
  <gloss>fondo</gloss>
```

```
  </etym>
</entry>
```

neume\'n(y)üm\ n [F, fr. ML pneuma, neuma, fr. Gk pneuma breath — more at **pneumatic**]: any of various symbols used in the notation of Gregorian chant … [WNC]

```
<entry>
  <etym>
    <lang>F</lang> fr. <lang>ML</lang>
    <mentioned>pneuma</mentioned>
    <mentioned>neuma</mentioned> fr. <lang>Gk</lang>
    <mentioned>pneuma</mentioned>
    <gloss>breath</gloss>
    <xr type="etym">more at <ptr target="#pneumatic"/>
    </xr>
  </etym>
  <def>any of various symbols … </def>
</entry>
<!-- ... -->
<entry xml:id="pneumatic">
<!-- ... -->
</entry>
```

9.3.5 Other Information

9.3.5.1 Examples

Dictionaries typically include examples of word use, usually accompanying definitions or translations. In some cases, the examples are quotations from another source, and are occasionally followed by a citation to the author.

The **<cit type="example">** element contains usage examples and associated information; the example text itself should be enclosed in a <q> or <quote> element. The <cit> element associates a quotation with a bibliographic reference to its source.

<q> (separated from the surrounding text with quotation marks) contains material which is marked as (ostensibly) being somehow different than the surrounding text, for any one of a variety of reasons including, but not limited to: direct speech or thought, technical terms or jargon, authorial distance, quotations from elsewhere, and passages that are mentioned but not used.

<quote> (quotation) contains a phrase or passage attributed by the narrator or author to some agency external to the text.

<cit> (cited quotation) contains a quotation from some other document, together with a bibliographic reference to its source. In a dictionary it may contain an example text with at least one occurrence of the word form, used in the sense being described, or a translation of the headword, or an example.

Examples frequently abbreviate the headword, and so their transcription will frequently make use of the <oRef> or <oVar> elements described below in section *9.4. Headword and Pronunciation References*.

Examples:

multiplex/…/ adj tech having many parts: the multiplex eye of the fly. LDOCE

```
<quote>the multiplex eye of the fly.</quote>
```

Or when one wants a more comprehensive representation of examples:

```
<cit  type="example">
  <quote>the multiplex eye of the fly.</quote>
</cit>
```

As the following example shows, <cit> can also contain elements such as <pron>, <def>, etc.

> **some** … 4. (*S*~ and *any* are used with *more*): Give me ~ more/s@'m0:(r)/OALD

```
<sense  n="4">
  <usg  type="colloc">
    <oRef  type="cap"/> and <mentioned>any</mentioned> are used with
  <mentioned>more</mentioned>
  </usg>
  <cit  type="example">
    <quote>Give me <oRef/> more</quote>
    <pron  extent="part">s@'m0:(r)</pron>
  </cit>
</sense>
```

In multilingual dictionaries, examples may also be accompanied by translations:

> **horrifier** … vt to horrify. **elle était horrifiée par la dépense** she was horrified at the expense. CR

```
<entry>
  <cit  type="translation"  xml:lang="en">
    <quote>to horrify</quote>
  </cit>
  <cit  type="example">
    <quote>elle était horrifiée par la dépense</quote>
    <cit  type="translation"  xml:lang="en">
      <quote>she was horrified at the expense.</quote>
    </cit>
  </cit>
</entry>
```

When a source is indicated, the example should be marked with a <cit> element:

> **valeur** … n. f. … 2. Vx. Vaillance, bravoure (spécial., au combat). ‘La valeur n'attend pas le nombre des années’ (Corneille). … DNT

```
<sense n="2">
  <usg type="time">Vx.</usg>
  <def>Vaillance, bravoure (spécial., au combat)</def>
  <cit type="example">
    <quote>La valeur n'attend pas le nombre des années</quote>
    <bibl>
      <author>Corneille</author>
    </bibl>
  </cit>
</sense>
```

9.3.5.2 Usage Information and Other Labels

Most dictionaries provide restrictive labels and phrases indicating the usage of given words or particular senses. Other phrases, not necessarily related to usage, may also be attached to forms, translations, cross-references, and examples. The following elements are provided to mark up such labels:

<usg> (usage) contains usage information in a dictionary entry.

<lbl> (label) contains a label for a form, example, translation, or other piece of information, e.g. abbreviation for, contraction of, literally, approximately, synonyms:, etc.

As indicated in the following section (*9.3.5.3. Cross-References to Other Entries*), the <lbl> element may be used for any kind of signicative phrase or label within the text. The <usg> element is a specialization of this to mark usage labels in particuar. Usage labels typically indicate

- temporal use (archaic, obsolete, etc.)
- register (slang, formal, taboo, ironic, facetious, etc.)
- style (literal, figurative, etc.)
- connotative effect (e.g. derogatory, offensive)
- subject field (Astronomy, Philosophy, etc.)
- national or regional use (Australian, U.S., Midland dialect, etc.)

Many dictionaries provide an explanation and/or a list of such usage labels in a preface or appendix. The type of the usage information may be indicated in the **type** attribute on the <usg> element. Some typical values are:

geo geographic area

time temporal, historical era ('archaic', 'old', etc.)

dom domain

reg register

style style (figurative, literal, etc.)

plev preference level ('chiefly', 'usually', etc.)

acc acceptability

lang language for foreign words, spellings pronunciations, etc.

gram grammatical usage

In addition to this kind of information, multilingual dictionaries often provide 'semantic cues' to help the user determine the right sense of a word in the source language (and hence the correct translation). These include

synonyms, concept subdivisions, typical subjects and objects, typical verb complements, etc. These labels may also be marked with the <usg> element; sample values for the **type** attribute in these cases include:

syn synonym given to show use

hyper hypernym given to show usage

colloc collocation given to show usage

comp typical complement

obj typical object

subj typical subject

verb typical verb

hint unclassifiable piece of information to guide sense choice

In this entry, one spelling is marked as geographically restricted:

> **colour** or U.S. **color** … CED

```
<form>
  <orth>colour</orth>
  <form>
    <usg type="geo">U.S.</usg>
    <orth>color</orth>
  </form>
</form>
```

In the next example, usage labels are used to indicate domains, register, and synonyms associated with different senses:

> **palette**[palEt] nf (a) (Peinture: lit, fig) palette. (b) (Boucherie) shoulder. (c) (aube de roue) paddle; (battoir à linge) beetle; (Manutention, Constr) pallet. CR

```
<sense n="a">
  <usg type="dom">Peinture</usg>
  <usg type="style">lit</usg>
  <usg type="style">fig</usg>
  <cit type="translation" xml:lang="en">
    <quote>palette</quote>
  </cit>
</sense>
<sense n="b">
  <usg type="dom">Boucherie</usg>
  <cit type="translation" xml:lang="en">
    <quote>shoulder</quote>
  </cit>
</sense>
<sense n="c">
  <sense>
    <usg type="syn">aube de roue</usg>
```

```
    <cit type="translation" xml:lang="en">
      <quote>paddle</quote>
    </cit>
  </sense>
  <sense>
    <usg type="syn">battoir à linge</usg>
    <cit type="translation" xml:lang="en">
      <quote>beetle</quote>
    </cit>
  </sense>
  <sense>
    <usg type="dom">Manutention</usg>
    <usg type="dom">Constr</usg>
    <cit type="translation" xml:lang="en">
      <quote>pallet</quote>
    </cit>
  </sense>
</sense>
```

When the usage label is hard to classify, it may be described as a 'hint':

rempaillage […] nm reseating, rebottoming (*with straw*). CR

```
<entry>
  <cit type="translation" xml:lang="en">
    <quote>reseating</quote>
    <quote>rebottoming</quote>
    <usg type="hint">with straw</usg>
  </cit>
</entry>
```

9.3.5.3 Cross-References to Other Entries

Dictionary entries frequently refer to information in other entries, often using extremely dense notations to convey the headword of the entry to be sought, the particular part of the entry being referred to, and the nature of the information to be sought there (synonyms, antonyms, usage notes, etymology, an illustration, etc.)

Cross-references may be tagged in dictionaries using the <ref> and <ptr> elements defined in the core module (section *3.6. Simple Links and Cross-References*). In addition, the <xr> element may be used to group all the information relating to a cross-reference.

<xr> (cross-reference phrase) contains a phrase, sentence, or icon referring the reader to some other location in this or another text.

<ref> (reference) defines a reference to another location, possibly modified by additional text or comment.

<ptr/> (pointer) defines a pointer to another location.

<lbl> (label) contains a label for a form, example, translation, or other piece of information, e.g. abbreviation for, contraction of, literally, approximately, synonyms:, etc.

As in other types of text, the actual pointing element (e.g. <ref> or <ptr>) is used to tag the cross-reference target proper (in dictionaries, usually the headword, possibly accompanied by a homograph number, a sense number, or other further restriction specifiying what portion of the target entry is being referred to). The <xr> element is used to group the target with any accompanying phrases or symbols used to label the cross-reference; the cross-reference label itself may be tagged as a <lbl> or may remain untagged. Both of the following are thus legitimate:

glee … Compare **madrigal** (sense 1) CED

```
<entry>
  <form>
    <orth>glee</orth>
  </form>
  <xr>Compare <ptr  target="#madrigal.1"/>
  </xr>
</entry>
<entry  xml:id="madrigal.1">
<!-- ...  -->
</entry>
```

hostellerie Syn. de hôtellerie (sens 1). DNT

```
<xr  type="syn">
  <lbl>Syn. de</lbl>
  <ref>hôtellerie (sens 1)</ref>.
</xr>
```

In addition to using, or not using, <lbl> to mark the cross-reference label, the two examples differ in another way. The former assumes that the first sense of *madrigal* has the identifier **madrigal.1**, and that the specific form of the reference in the source volume can be reconstructed, if needed, from that information. The latter does not require the first sense of 'hôtellerie' to have an identifier, and retains the print form of the cross-reference; by omitting the **target** attribute of the <ref> element, however, the second example does assume implicitly either that some software could usefully parse the phrase tagged as a <ref> and find the location referred to, or else that such processing will not be necessary.

The **type** attribute on the pointing element or on the <xr> element may be used to indicate what kind of cross-reference is being made, using any convenient typology. Since different dictionaries may label the same kind of cross-reference in different ways, it may be useful to give normalized indications in the**type** attribute, enabling the encoder to distinguish irregular forms of cross-reference more reliably:

rose2 … vb. the past tense of **rise**. CED

```
<entry  n="2">
  <form>
    <orth>rose</orth>
  </form>
  <xr  type="inflectedForm">
    <lbl>the past tense of</lbl>
    <ref  target="#rise">rise</ref>
  </xr>
</entry>
```

from cross-references for synonyms and the like:

antagonist … syn see **adverse** W7

```
<xr type="synonym">
  <lbl>syn see</lbl>
  <ref target="#adverse">adverse</ref>
</xr>
```

Strictly speaking, the reference above is not to the entry for *adverse*, but to the list of synonyms found at that entry. Slightly more complicated is the following reference to an illustration accompanying another entry:

> **ax, axe** … → see the illus at **tool** OALD

This entry refers to the illustration at the entry for *tool*, not the entry itself. The **target** attribute might give the identifier of the illustration itself, or of the enclosing entry (in which case the **type** attribute might be used to infer that the reference is actually to the illustration, not the entry as a whole).

```
<xr type="illustration">
  <lbl>see the illus at</lbl>
  <ptr target="#tool.illus"/>
</xr>
```

In some cases, the cross-reference is to a particular subset of the meanings of the entry in question:

> **globe** …V. **armillaire** (sphère) PR

```
<xr>V. <ref target="#armillaire">armillaire</ref>
  <lbl type="sense-restriction">sphère</lbl>
</xr>
```

Cross-references occasionally occur in definition texts, example texts, etc., or may be free-standing within an entry. These may typically be encoded using <ref> or <ptr>, without an enclosing <xr>. For example:

> **entacher** … *Acte entaché de nullité*, contenant un vice de forme ou passé par un incapable*. DNT

The asterisk signals a reference to the entry for *incapable*.

```
<def>contenant un vice de forme ou passé par un <ptr target="#incapable"/>.</def>
```

In some cases, the form in the definition is inflected, and thus <ref> must be used to indicate more exactly the intended target, as here:

> **justifier** …4. IMPRIM Donner a (une ligne) une longeur convenable au moyen de blancs (2, sens 1, 3). DNT

```
<sense n="4">
  <usg type="dom">imprim</usg>
  <def>Donner a (une ligne) une longeur convenable au moyen de
  <ref target="#blanc-2.1 #blanc-2.3">blancs (2, sens 1, 3)</ref>
  </def>
</sense>
```

9.3.5.4 Notes within Entries

Dictionaries may include extensive explanatory notes about usage, grammar, context, etc. within entries. Very often, such notes appear as a separate section at the end of an entry. The standard <note> element should be used for such material.

<note> contains a note or annotation.

For example:

> **ain't**(eInt)*Not standard. contraction of* am not, is not, are not, have not *or* has not: *I ain't seen it.* ….**Usage.** Although the interrogative form *ain't I?* would be a natural contraction of *am I not?*, it is generally avoided in spoken English and never used in formal English. CED

```
<entry>
  <form  type="contr">
    <orth>ain't</orth>
    <pron>eInt</pron>
  </form>
  <usg  type="reg">Not standard</usg>
  <form  type="full">
    <lbl>contraction of</lbl>
    <orth>am not</orth>
    <orth>is not</orth>
    <orth>are not</orth>
    <orth>have not</orth>
    <orth>has not</orth>
  </form>
  <cit  type="example">
    <quote>I ain't seen it.</quote>
  </cit>
  <note  type="usage">Although the interrogative form <mentioned>ain't
        I?</mentioned> would be a natural contraction of <mentioned>am I
        not?</mentioned>, it is generally avoided in spoken English and
     never used in formal English.</note>
</entry>
```

The formal declaration for <note> is given in section *3.8. Notes, Annotation, and Indexing*.

9.3.6 Related Entries

The <re> element encloses a degenerate entry which appears in the body of another entry for some purpose. Many dictionaries include related entries for direct derivatives or inflected forms of the entry word, or for compound words, phrases, collocations, and idioms containing the entry word.

Related entries can be complex, and may in fact include any of the information to be found in a regular entry. Therefore, the <re> element is defined to contain the same elements as an <entry> element, with the exception that it may not contain any nested <re> elements.

Examples:

> **bevvy**("bEvI) Dialect. ~ n., pl. -vies. 1. a drink, esp. an alcoholic one: we had a few bevvies last night. 2. a night of drinking. ~ vb. - vies, -vying, -vied (intr.) 3. to drink alcohol [probably from Old French bevee, buvee, drinking] —'bevvied adj. CED

```
<entry>
  <form>
    <orth>bevvy</orth>
    <pron>"bEvI</pron>
  </form>
  <usg  type="reg">Dialect</usg>
  <hom>
    <gramGrp>
      <pos>n</pos>
    </gramGrp>
    <sense  n="1">
      <def>a drink, esp. an alcoholic one: we had a few bevvies last night.</def>
    </sense>
  </hom>
<!-- ... sense 2 ... -->
  <hom>
    <gramGrp>
      <pos>vb</pos>
    </gramGrp>
    <sense  n="3">
      <def>to drink alcohol</def>
    </sense>
  </hom>
  <etym>probably from <lang>Old French</lang>
    <mentioned>bevee</mentioned>, <mentioned>buvee</mentioned>
    <gloss>drinking</gloss>
  </etym>
  <re  type="derived">
    <form>
      <orth>bevvied</orth>
    </form>
    <gramGrp>
      <pos>adj</pos>
    </gramGrp>
  </re>
</entry>
```

9.4 Headword and Pronunciation References

Examples, definitions, etymologies, and occasionally other elements such as cross-references, orthographic forms, etc., often contain a shortened or iconic reference to the headword, rather than repeating the headword itself. The references may be to the orthographic form or to the pronunciation, to the form given or to a variant of that form. The following elements are used to encode such iconic references to a headword:

<oRef/> (orthographic-form reference) in a dictionary example, indicates a reference to the orthographic form(s) of the headword.

@**type** indicates the kind of typographic modification made to the headword in the reference.

<pRef/> (pronunciation reference) in a dictionary example, indicates a reference to the pronunciation(s) of the headword.

<oVar> (orthographic-variant reference) in a dictionary example, indicates a reference to variant orthographic form(s) of the headword.

@**type** indicates the kind of variant involved.

<pVar> (pronunciation-variant reference) in a dictionary example, indicates a reference to variant pronunciation(s) of the headword.

As members of the class att.ptrLike.form, all these elements share a **target** attribute, which may optionally be used to resolve any ambiguity about the headword form being referred to.

att.ptrLike.form (form pointers) common attributes for elements in the dictionary base which point at orthographic or pronunciation forms of the headword.

@**target** identifies the orthographic form or pronunciation referred to.

Headword references come in a variety of formats:

~ indicates a reference to the full form of the headword

pref~ gives a prefix to be affixed to the headword

~suf gives a suffix to be affixed to the headword

A~ gives the first letter in uppercase, indicating that the headword is capitalized

pref~suf gives a prefix and a suffix to be affixed to the headword

a. gives the initial of the word followed by a full stop, to indicate reference to the full form of the headword

A. refers to a capitalized form of the headword

The <oRef> element should be used for iconic or shortened references to the orthographic form(s) of the headword itself. It is an empty element and replaces, rather than enclosing, the reference. Note that the reference to a headword is not necessarily a simple string replacement. In the example '**colour**1, (US = color) …~ films; ~ TV; Red, blue and yellow are ~s.'OALD, the tilde stands for either headword form (*colour*, *color*).

Examples:

colonel … army officer above a lieutenant-~. OALD

```
<def>army officer above a lieutenant-<oRef/>
</def>
```

academy … The Royal A~ of Arts OALD

```
<q>The Royal <oRef  type="cap"/> of Arts</q>
```

The following example demonstrates the use of the **target** attribute to refer to a specific form of the headword:

vag- or **vago-** comb form … : vagus nerve < **vag**al > < **vago**tomy > W7

```
<entry>
  <form>
    <orth  xml:id="di-o1">vag-</orth>
    <orth  xml:id="di-o2">vago-</orth>
  </form>
  <def>vagus nerve</def>
  <cit  type="example">
    <quote>
      <oRef  target="#di-o1"  type="nohyph"/>al</quote>
    <quote>
      <oRef  target="#di-o2"  type="nohyph"/>tomy</quote>
  </cit>
</entry>
```

In many cases the reference is not to the orthographic form of the headword, but rather to another form of the headword — usually to an inflected form. In these cases, the element <oVar> should be used; this element takes as its content the string as it appears in the text.

take … < Mr Burton **took** us for French > NPEG

```
<cit  type="example">
  <quote>Mr Burton <oVar  type="pt">took</oVar> us for French</quote>
</cit>
```

take … < was quite **~n** with him > NPEG

```
<cit  type="example">
  <quote>was quite <oVar  type="pp">
     <oRef/>n</oVar> with him</quote>
</cit>
```

The next example shows a discontinuous reference, using the attributes **next** and **prev**, which are defined in the additional module for linking, segmentation, and alignment (see chapter *16. Linking, Segmentation, and Alignment*) and therefore require that that module be selected in addition to that for dictionaries.

mix up… < it's easy to **mix** her **up** with her sister > NPEG

```
<cit  type="example">
  <quote>it's easy to <oVar  next="#ov2"  xml:id="ov1">mix</oVar>
    her <oVar  prev="#ov1"  xml:id="ov2">up</oVar> with her sister</quote>
</cit>
```

In addition, some dictionaries make reference to the pronunciation of the headword in the pronunciation of related entries, variants, or examples. The <pRef> and<pVar> elements should be used for such references.

hors d'oeuvre`/,aw'duhv` (Fr O:r dœvr)/ n, pl hors d'oeuvres also hors d'oeuvre `/'duhv(z)` (Fr ~)/ NPEG

```
<form>
  <orth>hors d'oeuvre</orth>
  <pron>%aU"dUv</pron>
  <form>
    <usg type="lang">Fr</usg>
    <pron xml:id="di-p2">OR d0vR</pron>
  </form>
</form>
<form type="infl">
  <number>pl</number>
  <orth>hors d'oeuvres</orth>
  <orth>hors d'oeuvre</orth>
  <pron extent="part">"dUv(z)</pron>
  <form>
    <usg type="lang">Fr</usg>
    <pron>
      <pRef target="#di-p2"/>
    </pron>
  </form>
</form>
```

Because headword and pronunciation references can occur virtually anywhere in an entry, the <oRef>, <oVar>, <pRef>, and <pVar> elements can appear within any other element defined for dictionary entries.

Since existing printed dictionaries use different conventions for headword references (swung dash, first letter abbreviated form, capitalization, or italicization of the word, etc.) the exact method used should be documented in the header.

9.5 Typographic and Lexical Information in Dictionary Data

Among the many possible views of dictionaries, it is useful to distinguish at least the following three, which help to clarify some issues raised with particular urgency by dictionaries, on account of the complexity of both their typography and their information structure.

- (a) the typographic view — the two-dimensional printed page, including information about line and page breaks and other features of layout
- (b) the editorial view — the one-dimensional sequence of tokens which can be seen as the input to the typesetting process; the wording and punctuation of the text and the sequencing of items are visible in this view, but specifics of the typographic realization are not
- (c) the lexical view — this view includes the underlying information represented in a dictionary, without concern for its exact textual form

For example, a domain indication in a dictionary entry might be broken over a line and therefore hyphenated ('naut-''ical'); the typographic view of the dictionary preserves this information. In a purely editorial view, the particular form in which the domain name is given in the particular dictionary (as 'nautical', rather than 'naut.', 'Naut.', etc.) would be preserved, but the fact of the line break would not. Font shifts might plausibly be included in either a strictly typographic or an editorial view. In the lexical view, the only information preserved concerning domain would be some standard symbol or string representing the nautical domain (e.g. 'naut.') regardless of the form in which it appears in the printed dictionary.

In practice, publishers begin with the lexical view — i.e., lexical data as it might appear in a database — and generate first the editorial view, which reflects editorial choices for a particular dictionary (such as the use of the abbreviation 'Naut.' for 'nautical', the fonts in which different types of information are to be rendered, etc.), and then the typographic view, which is tied to a specific printed rendering. Computational linguists and philologists often begin with the typographic view and analyse it to obtain the editorial and/or lexical views.

Some users may ultimately be concerned with retaining only the lexical view, or they may wish to preserve the typographic or editorial views as a reference text, perhaps as a guard against the loss or misinterpretation of information in the translation process. Some researchers may wish to retain all three views, and study their interrelations, since research questions may well span all three views.

In general, an electronic encoding of a text will allow the recovery of at least one view of that text (the one which guided the encoding); if editorial and typographic practices are consistently applied in the production of a printed dictionary, or if exceptions to the rules are consistently recorded in the electronic encoding, then it is *in principle* possible to recover the editorial view from an encoding of the lexical view, and the typographic view from an encoding of the editorial view. In practice, of course, the severe compression of information in dictionaries, the variety of methods by which this compression is achieved, the complexity of formulating completely explicit rules for editorial and typographic practice, and the relative rarity of complete consistency in the application of such rules, all make the mechanical transformation of information from one view into another something of a vexed question.

This section describes some principles which may be useful in capturing one or the other of these views as consistently and completely as possible, and describes some methods of attempting to capture more than one view in a single encoding. Only the editorial and lexical views are explicitly treated here; for methods of recording the physical or typographic details of a text, see chapter *11. Representation of Primary Sources*. Other approaches to these problems, such as the use of repetitive encoding and links to show their correspondences, or the use of feature structures to capture the information structure, and of the **ana** and **inst** attributes to link feature structures to a transcription of the editorial view of a dictionary, are not discussed here (for feature structures, see chapter *18. Feature Structures*. For linkage of textual form and underlying information, see chapter *17. Simple Analytic Mechanisms*).

9.5.1 Editorial View

Common practice in encoding texts of all sorts relies on principles such as the following, which can be used successfully to capture the editorial view when encoding a dictionary:

1. All characters of the source text should be retained, with the possible exception of rendition text (for which see further below).

2. Characters appearing in the source text should typically be given as character data content in the document, rather than as the value of an attribute; again, rendition text may optionally be excepted from this rule.

3. Apart from the characters or graphics in the source text, nothing else should appear as content in the document, although it may be given in attribute values.

4. The material in the source text should appear in the encoding in the same order. Complications of the character sequence by footnotes, marginal notes, etc., text wrapping around illustrations, etc., may be dealt with by the usual means (for notes, see section *3.8. Notes, Annotation, and Indexing*).[2]

In a very conservative transcription of the editorial view of a text, rendition characters (e.g. the commas, parentheses, etc., used in dictionary entries to signal boundaries among parts of the entry) and rendition text (for example, conjunctions joining alternate headwords, etc.) are typically retained. Removing the tags from such a transcription will leave all and only the characters of the source text, in their original sequence.[3]

[2] Complications of sequence caused by marginal or interlinear insertions and deletions, which are frequent in manuscripts, or by unconventional page layouts, as in concrete poetry, magazines with imaginative graphic designers, and texts about the nature of typography as a medium, typically do not occur in dictionaries, and so are not discussed here.

[3] This is a slight oversimplification. Even in conservative transcriptions, it is common to omit page numbers, signatures of gatherings, running titles and the like. The simple description above also elides, for the sake of simplicity, the difficulties of assigning a meaning to the phrase 'original sequence' when it is applied to the printed characters of a source text; the 'original sequence' retained or recovered from a conservative transcription of the editorial view is, of course, the one established during the transcription by the encoder.

Consider, for example, the following entry:

> **pinna** ('pIn@) n., pl. -nae (-ni:) *or* -nas. 1. any leaflet of a pinnate compound leaf. 2. *Zoology*. a feather, wing, fin, or similarly shaped part. 3. another name for **auricle** (sense 2). [C18: via New Latin from Latin: wing, feather, fin] CED

A conservative encoding of the editorial view of this entry, which retains all rendition text, might resemble the following:

```
<entry>
  <form>
    <orth>pinna</orth>
    <pron>("pIn@)</pron>
  </form>
  <gramGrp>
    <pos>n.</pos>, </gramGrp>
  <form  type="infl">
    <number>pl.</number>
    <form>
      <orth  type="lat"  extent="part">-nae</orth>
      <pron  extent="part">(-ni:)</pron>
    </form> or <orth  type="std"  extent="part">-nas</orth>
  </form>
  <sense  n="1">1. <def>any leaflet of a pinnate compound leaf.</def>
  </sense>
  <sense  n="2">2. <usg  type="dom">Zoology</usg>
    <def>a feather, wing, fin, or similarly shaped part.</def>
  </sense>
  <sense  n="3">3. <xr  type="syn">
     <lbl>another name for</lbl>
     <ref  target="#auricle.2">auricle (sense 2).</ref>
    </xr>
  </sense>
  <etym>[<date>C18</date>: via <lang>New Latin</lang> from <lang>Latin</lang>:
  <gloss>wing</gloss>, <gloss>feather</gloss>,
  <gloss>fin</gloss>]</etym>
</entry>
<entry  xml:id="auricle.2">
<!-- .... -->
</entry>
```

A somewhat simplified encoding of the editorial view of this entry might exploit the fact that rendition text is often systematically recoverable. For example, parentheses consistently appear around pronunciation in this dictionary, and thus are effectively implied by the start- and end-tags for <pron>.[4] In such an encoding, removing the tags should exactly reproduce the sequence of characters in the source, minus rendition text. The original character sequence can be recovered fully by replacing tags with any rendition text they imply.

Encoding in this way, the example given above might resemble the following. The <tagUsage> element in the header would be used to record the following patterns of rendition text:

- parentheses appear around <pron> elements
- commas appear before inflected forms
- the word 'or' appears before alternate forms

[4] The omission of rendition text is particularly common in systems for document production; it is considered good practice there, since automatic generation of rendition text is more reliable and more consistent than attempting to maintain it manually in the electronic text.

- brackets appear around the etymology
- full stops appear after <pos>, inflection information, and sense numbers
- senses are numbered in sequence unless otherwise specified using the global **n** attribute

```
<entry>
  <form>
    <orth>pinna</orth>
    <pron>"pIn@</pron>
  </form>
  <gramGrp>
    <pos>n</pos>
  </gramGrp>
  <form type="infl">
    <number>pl</number>
    <form>
      <orth type="lat" extent="part">-nae</orth>
      <pron extent="part">-ni:</pron>
    </form>
    <orth type="std" extent="part">-nas</orth>
  </form>
  <sense n="1">
    <def>any leaflet of a pinnate compound leaf.</def>
  </sense>
  <sense n="2">
    <usg type="dom">Zoology</usg>
    <def>a feather, wing, fin, or similarly shaped part.</def>
  </sense>
  <sense n="3">
    <xr type="syn">
      <lbl>another name for</lbl>
      <ref>auricle (sense 2).</ref>
    </xr>
  </sense>
  <etym>
    <date>C18</date>: via <lang>New Latin</lang> from <lang>Latin</lang>:
  <gloss>wing</gloss>, <gloss>feather</gloss>, <gloss>fin</gloss>
  </etym>
</entry>
```

When rendition text is omitted, it is recommended that the means to regenerate it be fully documented, using the <tagUsage> element of the TEI header.

If rendition text is used systematically in a dictionary, with only a few mistakes or exceptions, the global attribute **rend** may be used on any tag to flag exceptions to the normal treatment. The values of the **rend** attribute are not prescribed, but it can be used with values such as **no-comma**, **no-left-paren**, etc. Specific values can be documented using the <rendition> element in the TEI header.

In the following (imaginary) example, no left parenthesis precedes the pronunciation:

> **biryani** or **biriani** %bIrI"A:nI) any of a variety of Indian dishes … [from Urdu]

This irregularity can be recorded thus:

```
<entry>
  <form>
```

```
    <orth>biryani</orth>
    <orth>biriani</orth>
    <pron  rend="noleftparen">%bIrI"A:nI</pron>
  </form>
  <def>any of a variety of Indian dishes … </def>
  <etym>from <lang>Urdu</lang>
  </etym>
</entry>
```

9.5.2 Lexical View

If the text to be interchanged retains only the lexical view of the text, there may be no concern for the recoverability of the editorial (not to speak of the typographic) view of the text. However, it is strongly recommended that the TEI header be used to document fully the nature of all alterations to the original data, such as normalization of domain names, expansion of inflected forms, etc.

In an encoding of the lexical view of a text, there are degrees of departure from the original data: normalizing inconsistent forms like 'nautical', 'naut'., 'Naut'., etc., to 'nautical' is a relatively slight alteration; expansion of 'delay -ed -ing' to 'delay, delayed, delaying' is a more substantial departure. Still more severe is the rearranging of the order of information in entries; for example:

- reorganizing the order of elements in an entry to show their relationship, as in

  > **clem** (klEm) or clam vb. clems, clemming, clemmed or clams, clamming, clammed CED

  where in a strictly lexical view one might wish to group 'clem' and 'clam' with their respective inflected forms.

- splitting an entry into two separate entries, as in

  > **celi.bacy** /"selIb@sI/ n [U] state of living unmarried, esp as a religious obligation. celi.bate /"selIb@t/ n [C] unmarried person (esp a priest who has taken a vow not to marry). OALD

  For some purposes, this entry might usefully be split into an entry for 'celibacy' and a separate entry for 'celibate'.

An encoding which captures the lexical view of the example given in the previous section might look something like the following. In this encoding:

- abbreviated forms have been silently expanded
- some forms have been moved to allow related forms to be grouped together
- the part of speech information has been moved to allow all forms to be given together
- the cross-reference to 'auricle' has been simplified

```
<entry>
  <form>
    <orth>pinna</orth>
    <pron>"pIn@</pron>
    <form  type="infl">
      <number>pl</number>
      <form>
        <orth  type="lat">pinnae</orth>
        <pron>'pIni:</pron>
      </form>
      <orth  type="std">pinnas</orth>
```

```
    </form>
  </form>
  <gramGrp>
    <pos>n</pos>
  </gramGrp>
  <sense n="1">
    <def>any leaflet of a pinnate compound leaf.</def>
  </sense>
  <sense n="2">
    <usg type="dom">Zoology</usg>
    <def>a feather, wing, fin, or similarly shaped part.</def>
  </sense>
  <sense n="3">
    <xr type="syn">
      <ptr target="#auricle.2"/>
    </xr>
  </sense>
  <etym>
    <date>C18</date>: via <lang>New Latin</lang> from <lang>Latin</lang>:
  <gloss>wing</gloss>, <gloss>feather</gloss>, <gloss>fin</gloss>
  </etym>
</entry>
```

9.5.3 Retaining Both Views

It is sometimes desirable to retain both the lexical and the editorial view, in which case a potential conflict exists between the two. When there is a conflict between the encodings for the lexical and editorial views, the principles described in the following sections may be applied.

9.5.3.1 Using Attribute Values to Capture Alternate Views

If the order of the data is the same in both views, then both views may be captured by encoding one 'dominant' view in the character data content of the document, and encoding the other using attribute values on the appropriate elements. If all tags were to be removed, the remaining characters would be those of the dominant view of the text.

The attribute class att.lexicographic is used to provide attributes for use in encoding multiple views of the same dictionary entry. These attributes are available for use on all elements defined in this chapter when the base module for dictionaries is selected.

When the editorial view is dominant, the following attributes may be used to capture the lexical view:

att.lexicographic defines a set of global attributes available on elements in the base tag set for dictionaries.

- @**norm** (normalized) gives a normalized form of information given by the source text in a non-normalized form
- @**split** gives the list of split values for a merged form

When the lexical view is dominant, the following attributes may be used to record the editorial view:

att.lexicographic defines a set of global attributes available on elements in the base tag set for dictionaries.

- @**orig** (original) gives the original string or is the empty string when the element does not appear in the source text.
- @**mergedIn** gives a reference to another element, where the original appears as a merged form.

One attribute is useful in either view:

att.lexicographic defines a set of global attributes available on elements in the base tag set for dictionaries.

@**opt** (optional) indicates whether the element is optional or not

For example, if the source text had the domain label 'naut.', it might be encoded as follows. With the editorial view dominant:

```
<usg norm="nautical" type="dom">naut.</usg>
```

The lexical view of the same label would transcribe the normalized form as content of the <usg> element, the typographic form as an attribute value:

```
<usg orig="naut." type="dom">nautical</usg>
```

If the source text gives inflectional information for the verb *delay* as 'delay, -ed, -ing', it might usefully be expanded to 'delayed, delayed, delaying'. An encoding of the editorial view might take this form:

```
<form>
  <orth>delay</orth>
  <form type="infl">
    <orth norm="delayed" extent="part">-ed</orth>
    <tns norm="pst,pstp"/>
  </form>
  <form type="infl">
    <orth norm="delaying" extent="part">-ing</orth>
    <tns norm="prsp"/>
  </form>
</form>
```

Note the use of the <tns> tag with null content, to enable the representation of implicit information even though it has no print realization.

The lexical view might be encoded thus:

```
<form>
  <orth>delay</orth>
  <form type="infl">
    <orth orig="-ed">delayed</orth>
    <tns orig="">pst</tns>
    <tns orig="">pstp</tns>
  </form>
  <form type="infl">
    <orth orig="-ing">delaying</orth>
    <tns orig="">prsp</tns>
  </form>
</form>
```

A particular problem may be posed by the common practice of presenting two alternate forms of a word in a single string, by marking some parts of the word as optional in some forms. The following entry is for a word which can be spelled either 'thyrostimuline' or 'thyréostimuline':

thyr(é)ostimuline [tiR(e)ostimylin] …

With the editorial view dominant, this entry might begin thus:

```
<form>
  <orth  split="thyrostimuline, thyréostimuline">thyr(é)ostimuline</orth>
  <pron  split="tiRostimylin, tiReostimylin">tiR(e)ostimylin</pron>
</form>
```

With the lexical view dominant, however, two <orth> and two <pron> elements would be encoded, in order to disentangle the two forms; the **orig** attribute would be used to record the typographic presentation of the information in the source.

```
<form>
  <orth  xml:id="dic-o1"  orig="thyr(é)ostimuline">thyrostimuline</orth>
  <pron  xml:id="dic-p1"  orig="tiR(e)ostimylin">tiRostimylin</pron>
</form>
<form>
  <orth  mergedIn="#dic-o1">thyréostimuline</orth>
  <pron  mergedIn="#dic-p1">tiReostimylin</pron>
</form>
```

This example might also be encoded using the **opt** attribute combined with the attributes **next** and **prev** defined in chapter *16. Linking, Segmentation, and Alignment*.

```
<form>
  <orth  next="#dict-o2"  xml:id="dict-o1">thyr</orth>
  <orth
     next="#dict-o3"
     prev="#dict-o1"
     xml:id="dict-o2"
     opt="true">é</orth>
  <orth  prev="#dict-o2"  xml:id="dict-o3">ostimuline</orth>
  <pron  next="#dict-p2"  xml:id="dict-p1">tiR</pron>
  <pron
     next="#dict-p3"
     prev="#dict-p1"
     xml:id="dict-p2"
     opt="true">e</pron>
  <pron  prev="#dict-p2"  xml:id="dict-p3">ostimylin</pron>
</form>
```

Note that this transcription preserves both the lexical and editorial views in a single encoding. However, it has the disadvantage that the strings corresponding to entire words do not appear in the encoding uninterrupted, and therefore complex processing is required to retrieve them from the encoded text. The use of the **opt** attribute is recommended, however, when long spans of text are involved, or when the optional part contains embedded tags.

For example, the following gives two definitions in one text: 'picture drawn with coloured chalk made into crayons', and 'coloured chalk made into crayons':

> **pas.tel** /"pastl US: pa"stel/ n 1 (picture drawn with) coloured chalk made into crayons. 2…
> OALD

A simple encoding solution would be to leave the definition text unanalysed, but this might be felt inadequate since it does not show that there are two definitions. A possible alternative encoding would be:

```
<sense n="1">
  <def>coloured
     chalk made into crayons</def>
  <def>picture drawn with coloured chalk
     made into crayons</def>
</sense>
```

This transcribes some characters of the source text twice, however, which deviates from the usual practice. The following encoding records both the editorial and lexical views:

```
<sense n="1">
  <def next="#d2" xml:id="d1" opt="true">picture drawn
     with</def>
  <def prev="#d1" xml:id="d2">coloured chalk made into
     crayons</def>
</sense>
```

9.5.3.2 Recording Original Locations of Transposed Elements

The attributes described in the previous section are useful only when the order of material is the same in both the editorial and the lexical view. When the two views impose different orders on the data, the standard linking mechanisms may be used to show the original location of material transposed in an encoding of the lexical view.

If the original is only slightly modified, the <anchor> element may be used to mark the original location of the material, and the **location** attribute may be used on the lexical encoding of that material to indicate its original location(s). Like those in the preceding section, this attribute is defined for the attribute class att.lexicographic:

att.lexicographic defines a set of global attributes available on elements in the base tag set for dictionaries.

@**opt** (optional) indicates whether the element is optional or not

For example:

pinna`("pIn@)` n., pl. -nae (-ni:) or -nas. CED

```
<form>
  <orth>pinna</orth>
  <pron>'pIn@</pron>
  <anchor xml:id="p01"/>
  <form type="infl">
    <number>pl</number>
    <form>
      <orth extent="part">-nae</orth>
      <pron extent="part">-ni:</pron>
    </form>
    <orth extent="part">-nas</orth>
  </form>
</form>
<gramGrp>
  <pos location="#p01">n</pos>
</gramGrp>
```

9.6 Unstructured Entries

The content model for the <entry> element provides an entry structure suitable for many average dictionaries, as well as many regular entries in more exotic dictionaries. However, the structure of some dictionaries does not allow the restrictions imposed by the content model for <entry>. To handle these cases, the <entryFree> and <dictScrap> elements are provided to support much wider variation in entry structure. The <dictScrap> element offers less freedom, in that it can only contain phrase level elements, but it can itself appear at any point within a dictionary entry where any of the structural components of a dictionary entry are permitted. As such, it acts as a container for otherwise anomalous parts of an entry.

The <entryFree> element places no constraints at all upon the entry: any element defined in this chapter, as well as all the normal phrase-level and inter-level elements, can appear anywhere within it. With the <entryFree> element, the encoder is free to use any element anywhere, as well as to use or omit grouping elements such as <form>, <gramGrp>, etc.

The <entryFree> element allows the encoding of entries which violate the structure specified for the <entry> element. For example, in the following entry from a dictionary already in electronic form, it is necessary to include a <pron> element within a <def>. This is not permitted in the content model for <entry>, but it poses no problem in the <entryFree> element.

```
<ent
h="demigod"> <hwd>demi|god</hwd> <pr> <ph>"demIgQd</ph> </pr> <hps
ps="n"> <hsn> <def>one who is partly divine and partly human</def>
<def>(in Gk myth, etc) the son of a god and a mortal woman,
eg<cf>Hercules</cf> <pr> <ph>"h3:kjUli:z</ph> </pr> </def> </hsn>
</hps> </ent> <ref  target="#DIC-OALD">OALD</ref>
```

```
<entryFree>
  <form>
    <orth>demigod</orth>
    <hyph>demi|god</hyph>
    <pron>"demIgQd</pron>
  </form>
  <gramGrp>
    <pos>n</pos>
  </gramGrp>
  <def>one who is partly divine and partly human</def>
  <def>(in Gk myth, etc) the son of a god and a mortal woman, eg
  <mentioned>Hercules</mentioned>
  </def>
  <pron>"h3:kjUli:z</pron>
</entryFree>
```

The <entryFree> element also makes it possible to transcribe a dictionary using only phrase-level ('atomic') elements—that is, using no grouping elements at all. This can be desirable if the encoder wants a completely 'flat' view, with no indication of or commitment to the association of one element with another. The following encoding uses no grouping elements, and keeps all rendition text:

biryani or **biriani**`(%bIrI"A:nI)` any of a variety of Indian dishes…[from Urdu] CED

```
<entryFree>
  <orth>biryani</orth> or <orth>biriani</orth>
  <pron>(%bIrI"A:nI)</pron>
  <def>any of a variety of Indian dishes …</def>
  <etym>[from <lang>Urdu</lang>]</etym>
</entryFree>
```

Here is an alternative way of representing the same structure, this time using <dictScrap>:

```
<entry>
  <dictScrap>
    <orth>biryani</orth> or <orth>biriani</orth>
    <pron>(%bIrI"A:nI)</pron>
    <def>any of a variety of Indian dishes …</def>
    <etym>[from <lang>Urdu</lang>]</etym>
  </dictScrap>
</entry>
```

9.7 The Dictionary Module

The module defined in this chapter makes available the following components:

Module dictionaries: Printed dictionaries

- Elements defined: case colloc def dictScrap entry entryFree etym form gen gram gramGrp hom hyph iType lang lbl mood number oRef oVar orth pRef pVar per pos pron re sense stress subc superEntry syll tns usg xr
- Classes defined: att.entryLike att.lexicographic att.ptrLike.form model.entryLike model.formPart model.gramPart model.morphLike model.ptrLike.form

The selection and combination of modules to form a TEI schema is described in *1.2. Defining a TEI Schema*.

Chapter 10

Manuscript Description

10.1 Overview

This module[1] defines a special purpose element which can be used to provide detailed descriptive information about handwritten primary sources. Although originally developed to meet the needs of cataloguers and scholars working with medieval manuscripts in the European tradition, the scheme presented here is general enough that it can also be extended to other traditions and materials, and is potentially useful for any kind of inscribed artefact.

The scheme described here is also intended to accommodate the needs of many different classes of encoders. On the one hand, encoders may be engaged in retrospective conversion of existing detailed descriptions and catalogues into machine tractable form; on the other, they may be engaged in cataloguing *ex nihilo*, that is, creating new detailed descriptions for materials never before catalogued. Some may be primarily concerned to represent accurately the description itself, as opposed to the ideas and interpretations the description represents; others may have entirely opposite priorities. At one extreme, a project may simply wish to capture an existing catalogue in a form that can be displayed on the Web, and which can be searched for literal strings, or for such features such as titles, authors and dates; at the other, a project may wish to create, in highly structured and encoded form, a detailed database of information about the physical characteristics, history, interpretation, etc. of the material, able to support practitioners of quantitative codicology as well as librarians.

To cater for this diversity, here as elsewhere, these Guidelines propose a flexible approach, in which encoders must choose for themselves the degree of prescription appropriate to their needs, and are provided with a choice of encoding mechanisms to support those differing degrees.

10.2 The Manuscript Description Element

The <msDesc> element will normally appear within the <sourceDesc> element of the header of a TEI conformant document, where the document being encoded is a digital representation of some manuscript original, whether as an encoded transcription, as a collection of digital images (as described in *11.1. Digital Facsimiles*), or as some combination of the two. However, in cases where the document being encoded is essentially a collection of manuscript descriptions, the <msDesc> element may be used in the same way as the bibliographic elements (<bibl>, <biblFull>, and <biblStruct>) making up the TEI element class model.biblLike. These typically appear within the <listBibl> element.

<msDesc> (manuscript description) contains a description of a single identifiable manuscript.

[1] This chapter is based on the work of the European MASTER (Manuscript Access through Standards for Electronic Records) project, funded by the European Union from January 1999 to June 2001, and led by Peter Robinson, then at the Centre for Technology and the Arts at De Montfort University, Leicester (UK). Significant input also came from a TEI Workgroup headed by Consuelo W. Dutschke of the Rare Book and Manuscript Library, Columbia University (USA) and Ambrogio Piazzoni of the Biblioteca Apostolica Vaticana (IT) during 1998-2000.

The <msDesc> element has the following components, which provide more detailed information under a number of headings. Each of these component elements is further described in the remainder of this chapter.

<msIdentifier> (manuscript identifier) contains the information required to identify the manuscript being described.

<head> (heading) contains any type of heading, for example the title of a section, or the heading of a list, glossary, manuscript description, etc.

<msContents> (manuscript contents) describes the intellectual content of a manuscript or manuscript part, either as a series of paragraphs or as a series of structured manuscript items.

<physDesc> (physical description) contains a full physical description of a manuscript or manuscript part, optionally subdivided using more specialised elements from the model.physDescPart class.

<history> groups elements describing the full history of a manuscript or manuscript part.

<additional> groups additional information, combining bibliographic information about a manuscript, or surrogate copies of it with curatorial or administrative information.

<msPart> (manuscript part) contains information about an originally distinct manuscript or part of a manuscript, now forming part of a composite manuscript.

The first of these components, <msIdentifier>, is the only one which is mandatory; it is described in more detail in *10.4. The Manuscript Identifier* below. It is followed optionally by one or more <head> elements, each holding a brief heading (see *10.5. The Manuscript Heading*), and then either one or more paragraphs, marked up as a series of <p> elements, or one or more of the specialized elements <msContents> (*10.6. Intellectual Content*), <physDesc> (*10.7. Physical Description*), <history> (*10.7.4. History*), and <additional> (*10.7.5. Additional information*). These elements are all optional, but if used they must appear in the order given here. Finally, in the case of a composite manuscript, a full description may also contain one or more <msPart> elements (*10.7.6. Manuscript Parts*).

To demonstrate the use of this module, consider the following sample manuscript description, chosen more or less at random from the Bodleian Library's *Summary catalogue* ([135])

28843. In Latin, on parchment: written in more than one hand of the 13th cent. in England: $7\frac{1}{4} \times 5\frac{3}{8}$ in., i + 55 leaves, in double columns: with a few coloured capitals.

'Hic incipit Bruitus Anglie,' the De origine et gestis Regum Angliae of Geoffrey of Monmouth (Galfridus Monumetensis): *beg.* 'Cum mecum multa & de multis.'

On fol. 54^{v} very faint is 'Iste liber est fratris guillelmi de buria de . . . Roberti ordinis fratrum Pred[icatorum],' 14th cent. (?): 'hanauilla' is written at the foot of the page (15th cent.). Bought from the rev. W. D. Macray on March 17, 1863, for £1 10*s*.

Now MS. **Add. A. 61.**

Figure 10.1: Entry for Bodleian MS. Add. A. 61 in Madan et al. 1895-1953

The simplest way of digitizing this catalogue entry would simply be to key in the text, tagging the relevant parts of it which make up the mandatory <msIdentifier> element, as follows:

```
<msDesc>
  <msIdentifier>
```

```
    <settlement>Oxford</settlement>
    <repository>Bodleian Library</repository>
    <idno>MS. Add. A. 61</idno>
    <altIdentifier  type="SC">
      <idno>28843</idno>
    </altIdentifier>
  </msIdentifier>
  <p>In Latin, on parchment: written in more than one hand of the 13th
     cent. in England: 7¼ x 5⅜ in., i + 55 leaves, in double columns: with
     a few coloured capitals.</p>
  <p>'Hic incipit Bruitus Anglie,' the De origine et gestis Regum
     Angliae of Geoffrey of Monmouth (Galfridus Monumetensis: beg. 'Cum
     mecum multa & de multis.'</p>
  <p>On fol. 54v very faint is 'Iste liber est fratris guillelmi de
     buria de ... Roberti ordinis fratrum Pred[icatorum],' 14th cent. (?):
     'hanauilla' is written at the foot of the page (15th cent.). Bought
     from the rev. W. D. Macray on March 17, 1863, for £1 10s.</p>
</msDesc>
```

Source: [135]

With a suitable stylesheet, this encoding would be as readable as the original; it would not, however, be very useful for search purposes since only shelfmarks and other identifiers are distinguished. To improve on this, one might wrap the paragraphs in the appropriate special-purpose first-child-level elements of <msDesc> and add some of the phrase-level elements available when the manuscript description module is in use:

```
<msDesc>
  <msIdentifier>
    <settlement>Oxford</settlement>
    <repository>Bodleian Library</repository>
    <idno>MS. Add. A. 61</idno>
    <altIdentifier  type="SC">
      <idno>28843</idno>
    </altIdentifier>
  </msIdentifier>
  <msContents>
    <p>
      <quote>Hic incipit Bruitus Anglie,</quote> the
    <title>De origine et gestis Regum Angliae</title>
        of Geoffrey of Monmouth (Galfridus Monumetensis):
        beg. <quote>Cum mecum multa & de multis.</quote>
        In Latin.</p>
  </msContents>
  <physDesc>
    <p>
      <material>Parchment</material>: written in
       more than one hand: 7¼ x 5⅜ in., i + 55 leaves, in double
       columns: with a few coloured capitals.</p>
  </physDesc>
  <history>
    <p>Written in
    <origPlace>England</origPlace> in the
    <origDate>13th cent.</origDate> On fol. 54v very faint is
    <quote>Iste liber est fratris guillelmi de buria de ... Roberti
           ordinis fratrum Pred[icatorum],</quote> 14th cent. (?):
    <quote>hanauilla</quote> is written at the foot of the page
```

```
       (15th cent.). Bought from the rev. W. D. Macray on March 17, 1863, for
       £1 10s.</p>
  </history>
</msDesc>
```

Source: [135]

Note that in this version the text has been slightly reorganized, but no actual rewriting has been necessary. The encoding now allows the user to search for such features as title, material, and date and place of origin; it is also possible to distinguish quoted material from descriptive passages and to search within descriptions relating to a particular topic (for example, history as distinct from material).

This process could be continued further, restructuring the whole entry so as to take full advantage of many more of the encoding possibilities provided by the module described in this chapter:

```
<msDesc>
  <msIdentifier>
    <settlement>Oxford</settlement>
    <repository>Bodleian Library</repository>
    <idno>MS. Add. A. 61</idno>
    <altIdentifier type="SC">
      <idno>28843</idno>
    </altIdentifier>
  </msIdentifier>
  <msContents>
    <msItem>
      <author xml:lang="en">Geoffrey of Monmouth</author>
      <author xml:lang="la">Galfridus Monumetensis</author>
      <title type="uniform" xml:lang="la">De origine et
            gestis Regum Angliae</title>
      <rubric xml:lang="la">Hic incipit Bruitus Anglie</rubric>
      <incipit xml:lang="la">Cum mecum multa & de multis</incipit>
      <textLang mainLang="la">Latin</textLang>
    </msItem>
  </msContents>
  <physDesc>
    <objectDesc form="codex">
      <supportDesc material="perg">
        <support>
          <p>Parchment.</p>
        </support>
        <extent>i + 55 leaves
        <dimensions scope="all" type="leaf" unit="inch">
            <height>7¼</height>
            <width>5⅜</width>
          </dimensions>
        </extent>
      </supportDesc>
      <layoutDesc>
        <layout columns="2">
          <p>In double columns.</p>
        </layout>
      </layoutDesc>
    </objectDesc>
    <handDesc>
      <p>Written in more than one hand.</p>
    </handDesc>
```

```
    <decoDesc>
      <p>With a few coloured capitals.</p>
    </decoDesc>
  </physDesc>
  <history>
    <origin>
      <p>Written in <origPlace>England</origPlace> in the <origDate notAfter="1300" notBefore="1200">13th
cent.</origDate>
      </p>
    </origin>
    <provenance>
      <p>On fol. 54v very faint is
      <quote xml:lang="la">Iste liber est fratris guillelmi de buria de <gap/>
               Roberti ordinis fratrum
               Pred<ex>icatorum</ex>
        </quote>, 14th cent. (?):
      <quote>hanauilla</quote> is written at the foot of the page
           (15th cent.).</p>
    </provenance>
    <acquisition>
      <p>Bought from the rev. <name key="MCRAYWD">W. D. Macray</name> on
      <date when="1863-03-17">March 17, 1863</date>, for £1 10s.</p>
    </acquisition>
  </history>
</msDesc>
```

Source: [135]

In the remainder of this chapter we discuss all of the encoding features demonstrated above, together with many other related matters.

10.3 Phrase-level Elements

When the msdescription module is in use, several extra elements are added to the phrase level class, and thus become available within paragraphs and elsewhere in the document. These elements are listed below in alphabetical order:

<catchwords> describes the system used to ensure correct ordering of the quires making up a codex or incunable, typically by means of annotations at the foot of the page.

<dimensions> contains a dimensional specification.

<heraldry> contains a heraldic formula or phrase, typically found as part of a blazon, coat of arms, etc.

<locus> defines a location within a manuscript or manuscript part, usually as a (possibly discontinuous) sequence of folio references.

<material> contains a word or phrase describing the material of which a manuscript (or part of a manuscript) is composed.

<watermark> contains a word or phrase describing a watermark or similar device.

<origDate> (origin date) contains any form of date, used to identify the date of origin for a manuscript or manuscript part.

<origPlace> (origin place) contains any form of place name, used to identify the place of origin for a manuscript or manuscript part.

<secFol> (second folio) The word or words taken from a fixed point in a codex (typically the beginning of the second leaf) in order to provide a unique identifier for it.

<signatures> contains discussion of the leaf or quire signatures found within a codex.

Within a manuscript description, many other standard TEI phrase level elements are available, notably those described in the Core module (*3. Elements Available in All TEI Documents*). Additional elements of particular relevance to manuscript description, such as those for names and dates, may also be made available by including the relevant module in one's schema.

10.3.1 Origination

The following elements may be used to provide information about the origins of any aspect of a manuscript:

<origDate> (origin date) contains any form of date, used to identify the date of origin for a manuscript or manuscript part.

<origPlace> (origin place) contains any form of place name, used to identify the place of origin for a manuscript or manuscript part.

The <origDate> and <origPlace> elements are specialized forms of the existing <date> and <name> elements respectively, used to indicate specifically the date and place of origin of a manuscript or manuscript part. Such information would normally be encoded within the <history> element, discussed in section *10.7.4. History*. <origDate> and <origPlace> can also be used to identify the place or date of origin of any aspect of the manuscript, such as its decoration or binding, when these are not of the same date as the manuscript itself. Both these elements are members of the att.editLike class, from which they inherit the following attributes:

att.editLike provides attributes describing the nature of a encoded scholarly intervention or interpretation of any kind.

The <origDate> element is a member of the att.datable class, and may thus also carry the following attributes:

att.datable provides attributes for normalization of elements that contain dates, times, or datable events.

10.3.2 Material

The <material> element can be used to tag any specific term used for the physical material of which a manuscript (or binding, seal, etc.) is composed.

<material> contains a word or phrase describing the material of which a manuscript (or part of a manuscript) is composed.

The element may appear wherever a term regarded as significant by the encoder occurs, as in the following example:

```
<binding>
  <p>Brown <material>calfskin</material>, previously with two clasps.</p>
</binding>
```

10.3.3 Watermarks and Stamps

Two further elements are provided to mark up other decorative features characteristic of manuscript leaves and bindings:

<watermark> contains a word or phrase describing a watermark or similar device.

<stamp> contains a word or phrase describing a stamp or similar device.

These element may appear wherever a term regarded as significant by the encoder occurs. The <watermark> element is most likely to be of use within the <support> element discussed in *10.7.1.1. Support* below. We give a simple example here:

```
<support>
  <material>Rag
    paper</material> with <watermark>anchor</watermark>
watermark
</support>
```

The <stamp> element will typically appear when text from the source is being transcribed, for example within a rubric in the following case:

```
<rubric>Apologyticu TTVLLIANI AC IGNORATIA IN XPO IHV<lb/>
SI NON LICET<lb/>
NOBIS RO<lb/>
manii imperii <stamp>Bodleian stamp</stamp>
  <lb/>
</rubric>
```

It may also appear as part of the detailed description of a binding:

```
<binding>
  <p>Modern calf recasing with original armorial stamp <stamp>Ex
       Bibliotheca J. Richard D.M.</stamp>
  </p>
</binding>
```

10.3.4 Dimensions

The <dimensions> element can be used to specify the size of some aspect of the manuscript, and thus may be thought of as a specialized form of the existing TEI <measure> element.

<dimensions> contains a dimensional specification.

@**type** indicates which aspect of the object is being measured.

The <dimensions> element will normally occur within the element describing the particular feature or aspect of a manuscript whose dimensions are being given; thus the size of the leaves would be specified within the <support> or <extent> element (part of the <physDesc> element discussed in *10.7.1. Object Description*), while the dimensions of other specific parts of a manuscript, such as accompanying materials, binding, etc., would be given in other parts of the description, as appropriate.

The following three elements are available within the <dimensions> element:

<height> contains a measurement measured along the axis parallel to the spine.

<width> contains a measurement measured along the axis perpendicular to the spine.

<depth> specifies a length measured across the spine.

These three elements, as well as <dimensions> itself, are all members of the att.dimensions class, and thus all carry the following attributes:

att.dimensions provides attributes for describing the size of physical objects.

@**unit** names the unit used for the measurement

@**scope** specifies the applicability of this measurement, where more than one object is being measured.

@**quantity** specifies the length in the units specified

@**precision** characterizes the precision of the specified measurement.

Each of these elements, if present, must be given in the order specified. The <dimensions> element may be repeated as often as necessary, with appropriate attribute values to indicate the nature and scope of the measurement concerned. For example, in the following case the leaf size and ruled space of the leaves of the manuscript are specified:

```
<dimensions type="leaves" unit="mm">
  <height scope="range">157-160</height>
  <width>105</width>
</dimensions>
<dimensions type="ruled" unit="mm">
  <height scope="most">90</height>
  <width scope="most">48</width>
</dimensions>
```

This indicates that for most leaves of the manuscript being described the ruled space is 90 mm high and 48 mm wide, while the leaves throughout are between 157 and 160 mm in height and 105 mm in width.

10.3.5 References to Locations within a Manuscript

The <locus> element is a specialized form of the <ref> element.

<locus> defines a location within a manuscript or manuscript part, usually as a (possibly discontinuous) sequence of folio references.

@**from** specifies the starting point of the location in a normalized form.

@**to** specifies the end-point of the location in a normalized form.

@**scheme** identifies the foliation scheme in terms of which the location is being specified.

The <locus> element is used to specify the location in the manuscript occupied by the element within which it appears. If, for example, it is used as the first component of a <msItem> or <msItemStruct> element, or of any of the more specific elements appearing within one (see further section *10.6. Intellectual Content* below) then it is understood to specify the location of that item within the manuscript being described.

A <locus> element can be used to identify any reference to one or more folios within a manuscript, wherever such a reference is appropriate. Locations are conventionally specified as a sequence of folio or page numbers, but may also be a discontinuous list, or a combination of the two. This specification should be given as the content of the <locus> element, using the conventions appropriate to the individual scholar or holding institution, as in the following example:

```
<msItem n="1">
  <locus>ff. 1-24r</locus>
  <title>Apocalypsis beati Ioannis Apostoli</title>
</msItem>
```

A normalized form of the location can also be supplied, using special purpose attributes on the <locus> element, as in the following revision of the above example:

```
<msItem n="1">
  <locus from="1r" to="24r">ff. 1-24r</locus>
  <title>Apocalypsis beati Ioannis Apostoli</title>
</msItem>
```

When the trans module described in chapter *11. Representation of Primary Sources* is included in a schema, an additional attribute **facs** is available, which may be used to associate a <locus> element with a digitized image, as in the following example:

```
<decoDesc>
  <p>Several of the miniatures in this section have been damaged and
     overpainted at a later date (e.g. the figure of Christ on
  <locus
       facs="http://www.example.com/images.fr#F33R">fol. 33r</locus>; the face of the
     Shepherdess on <locus
       facs="http://www.example.com/images.fr#F59V">fol. 59v</locus>,
     etc.).</p>
</decoDesc>
```

Here, the **facs** attribute uses a URI reference to point directly to images of the relevant pages, held presumably in an image archive not forming part of the resource containing the manuscript description in question. In the case where such images do form part of a single virtual document, along with transcriptions and metadata, the methods described in *11.1. Digital Facsimiles* should be used in preference.

Where a transcription of the relevant pages is also available, this may be pointed to using the **target** attribute, as in the following example:

```
<!-- within ms description --><msItem  n="1">
  <locus  target="#f1r #f1v #f2r">ff. 1r-2r</locus>
  <author>Ben Jonson</author>
  <title>Ode to himself</title>
  <rubric  rend="italics"> An Ode<lb/> to him selfe.</rubric>
  <incipit>Com leaue the loathed stage</incipit>
  <explicit>And see his chariot triumph ore his wayne.</explicit>
  <bibl>
    <name>Beal</name>, <title>Index 1450-1625</title>, JnB 380</bibl>
</msItem>
<!-- within transcription ... -->
<pb  xml:id="f1r"/>
<!-- ... -->
<pb  xml:id="f1v"/>
<!-- ... -->
<pb  xml:id="f2r"/>
<!-- ... -->
```

Where a manuscript contains more than one foliation, the **scheme** attribute may be used to distinguish them. For example, MS 65 Corpus Christi College, Cambridge contains two fly leaves bearing music. These leaves have modern foliation 135 and 136 respectively, but are also marked with an older foliation. This may be preserved in an encoding such as the following:

```
<locus  scheme="#original">XCIII</locus>
<locus  scheme="#modern">135</locus>
```

Here the **scheme** attribute points to a <foliation> element providing more details about the scheme used, as further discussed in *10.7.1.4. Foliation* below.

10.3.6 Names of Persons, Places, and Organizations

The standard TEI element <name> may be used to identify names of any kind occurring within a description:

<name> (name, proper noun) contains a proper noun or noun phrase.

@**type** indicates the type of the object which is being named by the phrase.

As further discussed in *3.5.1. Referring Strings*, this element is a member of the TEI class att.naming, from which it inherits the following attributes:

att.naming provides attributes common to elements which refer to named persons, places, organizations etc.

- @**key** provides an external means of locating a full definition for the entity (or entities) being named, such as a database record key or other token.
- @**ref** (reference) provides an explicit means of locating a full definition for the entity being named by means of a URI.

Here are some examples of the use of the <name> element:

```
<name type="person">Thomas Hoccleve</name>
<name type="place">Villingaholt</name>
<name type="org">Vetus Latina Institut</name>
<name type="person" ref="#HOC001">Occleve</name>
```

Note that the <name> element is defined as providing information about a *name*, not the person, place, or organization to which that name refers. In the last example above, the **ref** attribute is used to associate the name with a more detailed description of the person named. This is provided by means of the <person> element, which becomes available when the namesdates module described in chapter *13. Names, Dates, People, and Places* is included in a schema. An element such as the following might then be used to provide detailed information about the person indicated by the name:

```
<person xml:id="HOC001">
  <persName>
    <surname>Hoccleve</surname>
    <forename>Thomas</forename>
  </persName>
  <birth notBefore="1368"/>
  <occupation>poet</occupation>
<!-- other personal data -->
</person>
```

Note that an instance of the <person> element must be provided for each distinct **ref** value specified. In the example above, the value **HOC001** must be found as the **xml:id** attribute of some <person>; the same value will be used as the **ref** attribute of every reference to Hoccleve in the document (however spelled), but there will only be one <person> element with this identifier.

Alternatively, the **key** attribute may be used to supply an unique identifying code for the person referenced by the name independently of both the existence of a <person> element and the use of the standard URI reference mechanism. If, for example, a project maintains as its authority file some non-digital resource, or uses a database which cannot readily be integrated with other digital resources for this purpose, the unique codes used by such 'offline' resources may be used as values for the **key** attribute. Although such practices clearly reduce the interchangeability of the resulting encoded texts, they may be judged more convenient or practical in certain situations.

All the <person> elements referenced by a particular document set should be collected together within a <listPerson> element, located in the TEI Header. This functions as a kind of prosopography for all the people referenced by the set of manuscripts being described, in much the same way as a <listBibl> element in the back matter may be used to hold bibliographic information for all the works referenced.

When the namesdates module described in chapter *13. Names, Dates, People, and Places* is included in a schema, similar mechanisms are used to maintain and reference canonical lists of places or organizations, as further discussed in sections *13.2.3. Place Names* and *13.2.2. Organizational Names* respectively.

10.3.7 Catchwords, Signatures, Secundo Folio

The <catchwords> element is used to describe one method by which correct ordering of the quires of a codex is ensured. Typically, this takes the form of a word or phrase written in the lower margin of the last leaf verso of a gathering, which provides a preview of the first recto leaf of the successive gathering. This may be a simple phrase such as the following:

```
<catchwords>Quires signed on the last leaf verso in roman numerals.</catchwords>
```

Alternatively, it may contain more details:

```
<catchwords>Vertical catchwords in the hand of the scribe placed along
the inner bounding line, reading from top to bottom.</catchwords>
```

The 'Signatures' element is used, in a similar way, to describe a similar system in which quires or leaves are marked progressively in order to facilitate arrangement during binding. For example:

```
<signatures>At the bottom of the first four leaves of quires 1-14 are
the remains of a series of quire signatures a-o plus roman figures in
a cursive hand of the fourteenth century.</signatures>
```

The <signatures> element can be used for either leaf signatures, or a combination of quire and leaf signatures, whether the marking is alphabetic, alphanumeric, or some ad hoc system, as in the following more complex example:

```
<signatures>Quire and leaf signatures in letters, [b]-v, and roman numerals;
those in quires 10 (1) and 17 (s) in red ink and different from others;
every third quire also signed with red crayon in arabic numerals in the
centre lower margin of the first leaf recto: "2" for quire 4 (f. 19),
"3" for quire 7 (f. 43); "4", barely visible, for quire 10 (f. 65), "5",
in a later hand, for quire 13 (f. 89), "6", in a later hand, for quire
16 (f. 113).</signatures>
```

The <secFol> element (for 'secundo folio') is used to record an identifying phrase (also called *dictio probatoria*) taken from a specific known point in a codex (for example the first few words on the second leaf). Since these words will differ from one copy of a text to another, the practice originated in the middle ages of using them when cataloguing a manuscript in order to distinguish individual copies of a work in a way which its opening words could not.

```
<secFol>(ando-)ssene in una villa</secFol>
```

10.3.8 Heraldry

Descriptions of heraldic arms, supporters, devices, and mottos may appear at various points in the description of a manuscript, usually in the context of ownership information, binding descriptions, or detailed accounts of illustrations. A full description may also contain a detailed account of the heraldic components of a manuscript independently considered. Frequently, however, heraldic descriptions will be cited as short phrases within other parts of the record. The phrase level element <heraldry> is provided to allow such phrases to be marked for further analysis, as in the following examples:

```
<p>Ownership stamp (xvii cent.) on i recto with the arms <heraldry>A bull
    passant within a bordure bezanty, in chief a crescent for difference</heraldry>
[Cole], crest, and the legend <quote>Cole Deum</quote>.</p>
<!-- ... -->
<p>A c. 8r fregio su due lati, <heraldry>stemma e imprese medicee</heraldry>
racchiudono l'inizio dell'epistolario di Paolino.</p>
```

10.4 The Manuscript Identifier

The <msIdentifier> element is intended to provide an unambiguous means of uniquely identifying a particular manuscript. This may be done in a structured way, by providing information about the holding institution and the call number, shelfmark, or other identifier used to indicate its location within that institution. Alternatively, or in addition, a manuscript may be identified simply by a commonly used name.

<msIdentifier> (manuscript identifier) contains the information required to identify the manuscript being described.

A manuscript's actual physical location may occasionally be different from its place of ownership; at Cambridge University, for example, manuscripts owned by various colleges are kept in the central University Library. Normally, it is the ownership of the manuscript which should be specified in the manuscript identifier, while additional or more precise information on the physical location of the manuscript can be given within the <adminInfo> element, discussed in section *10.7.5.1. Administrative information* below.

The following elements are available within <msIdentifier> to identify the holding institution:

<country> contains the name of a geo-political unit, such as a nation, country, colony, or commonwealth, larger than or administratively superior to a region and smaller than a bloc.

<region> contains the name of an administrative unit such as a state, province, or county, larger than a settlement, but smaller than a country.

<settlement> contains the name of a settlement such as a city, town, or village identified as a single geo-political or administrative unit.

<institution> contains the name of an organization such as a university or library, with which a manuscript is identified, generally its holding institution.

<repository> contains the name of a repository within which manuscripts are stored, possibly forming part of an institution.

These elements are all structurally equivalent to the standard TEI <name> element with an appropriate value for its **type** attribute; however the use of this 'syntactic sugar' enables the model for <msIdentifier> to be constrained rather more tightly than would otherwise be possible. Specifically, only one of each of the elements listed above may appear within the <msIdentifier> and they must, if present, appear in the order given.

These elements are all also members of the attribute class att.naming, from which they inherit the following attribute:

att.naming provides attributes common to elements which refer to named persons, places, organizations etc.

The following elements are used within <msIdentifier> to provide different ways of identifying the manuscript within its holding institution:

<collection> contains the name of a collection of manuscripts, not necessarily located within a single repository.

<idno> (identifying number) supplies any standard or non-standard number used to identify a bibliographic item.

<altIdentifier> (alternative identifier) contains an alternative or former structured identifier used for a manuscript, such as a former catalogue number.

<msName> (alternative name) contains any form of unstructured alternative name used for a manuscript, such as an 'ocellus nominum', or nickname.

Major manuscript repositories will usually have a preferred form of citation for manuscript shelfmarks, including rules about punctuation, spacing, abbreviation, etc., which should be adhered to. Where such a format also contains information which might additionally be supplied as a distinct subcomponent of the <msIdentifier>, for example a collection name, a decision must be taken as to whether to use the more specific element, or to include such information within the <idno> element. For example, the manuscript formally identified as 'El 26 C 0' forms a part of the Ellesmere ('El') collection. Either of the following encodings is therefore feasible:

```
<msIdentifier>
  <country>USA</country>
  <region>California</region>
  <settlement>San Marino</settlement>
  <repository>Huntington Library</repository>
  <collection>El</collection>
  <idno>26 C 9</idno>
  <msName>The Ellesmere Chaucer</msName>
</msIdentifier>
```

```
<msIdentifier>
  <country>USA</country>
  <region>California</region>
  <settlement>San Marino</settlement>
  <repository>Huntington Library</repository>
  <idno>El 26 C 9</idno>
  <msName>The Ellesmere Chaucer</msName>
</msIdentifier>
```

In the former example, the preferred form of the identifier can be retrieved by prefixing the content of the <idno> element with that of the <collection> element, while in the latter it is given explicitly. The advantage of the former is that it it simplifies accurate retrieval of all manuscripts from a given collection; the disadvantage is that encoded abbreviations of this kind may not be as immediately comprehensible. Care should be taken to avoid redundancy: for example

```
<collection>El</collection>
<idno>El 26 C 9</idno>
```

would clearly be inappropriate. Equally clearly,

```
<collection>Ellesmere</collection>
<idno>El 26 C 9</idno>
```

might be considered helpful in some circumstances (if, for example, some of the items in the Ellsemere collection had shelfmarks which did not begin 'El')

In cases where the shelfmark contains no information about the collection, it may be necessary to provide this explicitly, as in the following example:

```
<msIdentifier>
  <country>USA</country>
  <region>New Jersey</region>
  <settlement>Princeton</settlement>
  <repository>Princeton University Library</repository>
  <collection>Scheide Library</collection>
  <idno>MS 71</idno>
  <msName>Blickling Homiliary</msName>
</msIdentifier>
```

In these examples, <msName> has been used to provide a common name other than the shelfmark by which a manuscript is known. Where a manuscript has several such names, more than one of these elements may be used, as in the following example:

```
<msIdentifier>
  <country>Danmark</country>
  <settlement>København</settlement>
  <repository>Det Arnamagnæanske Institut</repository>
  <idno>AM 45 fol.</idno>
  <msName  xml:lang="la">Codex Frisianus</msName>
  <msName  xml:lang="is">Fríssbók</msName>
</msIdentifier>
```

Here the globally available **xml:lang** attribute has been used to specify the language of the alternative names.

In very rare cases a repository may have only one manuscript (or only one of any significance), which will have no shelfmark as such but will be known by a particular name or names. In such circumstances, the <idno> element may be omitted, and the manuscript identified by the name or names used for it, using one or more <msName> elements, as in the following example:

```
<msIdentifier>
  <settlement>Rossano</settlement>
  <repository  xml:lang="it">Biblioteca arcivescovile</repository>
  <msName  xml:lang="la">Codex Rossanensis</msName>
  <msName  xml:lang="la">Codex purpureus</msName>
  <msName  xml:lang="en">The Rossano Gospels</msName>
</msIdentifier>
```

Where manuscripts have moved from one institution to another, or even within the same institution, they may have identifiers additional to the ones currently used, such as former shelfmarks, which are sometimes retained even after they have been officially superseded. In such cases it may be useful to supply an alternative identifier, with a detailed structure similar to that of the <msIdentifier> itself. The following example shows a manuscript which had shelfmark `II-M-5` in the collection of the Duque de Osuna, but which now has the shelfmark `MS 10237` in the National Library in Madrid:

```
<msIdentifier>
  <settlement>Madrid</settlement>
  <repository>Biblioteca Nacional</repository>
  <idno>MS 10237</idno>
  <altIdentifier>
    <region>Andalucia</region>
    <settlement>Osuna</settlement>
    <repository>Duque de Osuna</repository>
    <idno>II-M-5</idno>
  </altIdentifier>
</msIdentifier>
```

Normally, such information would be dealt with under <history>, except in cases where a manuscript is likely still to be referred to or known by its former identifier. For example, an institution may have changed its call number system but still wish to retain a record of the earlier number, perhaps because the manuscript concerned is frequently cited in print under its previous number:

```
<msIdentifier>
  <settlement>Berkeley</settlement>
  <institution>University of California</institution>
  <repository>Bancroft Library</repository>
  <idno>UCB 16</idno>
  <altIdentifier>
    <idno>2MS BS1145 I8</idno>
  </altIdentifier>
</msIdentifier>
```

Where (as in this example) no repository is specified for the <altIdentifier>, it is assumed to be the same as that of the parent <msIdentifier>. Where the holding institution has only one preferred form of citation but wishes to retain the other for internal administrative purposes, the secondary could be given within <altIdentifier> with an appropriate value on the **type** attribute:

```
<msIdentifier>
  <settlement>Oxford</settlement>
  <repository>Bodleian Library</repository>
  <idno>MS. Bodley 406</idno>
  <altIdentifier  type="SC">
    <idno>2297</idno>
  </altIdentifier>
</msIdentifier>
```

It might, however, be preferable to include such information within the <adminInfo> element discussed in section *10.7.5.1. Administrative information* below.

Cases of such changed or alternative identifiers should be clearly distinguished from cases of 'scattered' manuscripts, that is to say manuscripts which although physically disjoint are nevertheless generally treated as single units. One well-known example is the Old Church Slavonic manuscript known as *Codex Suprasliensis*, substantial parts of which are to be found in three separate repositories, in Ljubljana, Warsaw, and St. Petersburg. This should be represented using three distinct <altIdentifier> elements, using an appropriate value on the type attribute to indicate that these three identifiers are not alternate ways of referring to the same physical object, but three parts of the same entity.

```
<msIdentifier>
  <msName  xml:lang="la">Codex Suprasliensis</msName>
  <altIdentifier  type="partial">
    <settlement>Ljubljana</settlement>
    <repository>Narodna in univerzitetna knjiznica</repository>
    <idno>MS Kopitar 2</idno>
    <note>Contains ff.  10 to 42 only</note>
  </altIdentifier>
  <altIdentifier  type="partial">
    <settlement>Warszawa</settlement>
    <repository>Biblioteka Narodowa</repository>
    <idno>BO 3.201</idno>
  </altIdentifier>
  <altIdentifier  type="partial">
    <settlement>Sankt-Peterburg</settlement>
    <repository>Rossiiskaia natsional'naia biblioteka</repository>
    <idno>Q.p.I.72</idno>
  </altIdentifier>
</msIdentifier>
```

As mentioned above, the smallest possible description is one that contains only the element <msIdentifier>; good practice in all but exceptional circumstances requires the presence within it of the three sub-elements <settlement>, <repository>, and <idno>, since they provide what is, by common consent, the minimum amount of information necessary to identify a manuscript.

10.5 The Manuscript Heading

Historically, the briefest possible meaningful description of a manuscript consists of no more than a title, e.g. *Polychronicon*. This will often have been enough to identify a manuscript in a small collection because the identity of the author is implicit. Where a title does not imply the author, and is thus insufficient to identify the main text of a manuscript, the author should be stated explicitly (e.g. *Augustinus, Sermones* or *Cicero, Letters*). Many inventories of manuscripts consist of no more than an author and title, with some form of copy-specific identifier, such as a shelfmark or 'secundo folio' reference (e.g. *Arch. B. 3. 2: Evangelium Matthei cum glossa*, *126. Isidori Originum libri octo*, *Biblia Hieronimi, 2o fo. opus est*); information on date and place of writing will sometimes also be included. The standard TEI element <head> element can be used to provide a brief description of this kind.

<head> (heading) contains any type of heading, for example the title of a section, or the heading of a list, glossary, manuscript description, etc.

In this way the cataloguer or scholar can supply in one place a minimum of essential information, such as might be displayed or printed as the heading of a full description. For example:

```
<head>Marsilius de Inghen, Abbreviata phisicorum Aristotelis; Italy, 1463.</head>
```

Any phrase-level elements, such as <title>, <name>, <date>, or the specialized elements <origPlace> and <origDate>, can also be used within a <head> element, but it should be remembered that the <head> element is intended principally to contain a heading. More structured information concerning the contents, physical form, or history of the manuscript should be given within the specialized elements described below, <msContents>, <physDesc>, <history>, etc. However, in simple cases, the <p> element may also be used to supply an unstructured collection of such information, as in the example given above (*10.2. The Manuscript Description Element*.

10.6 Intellectual Content

The <msContents> element is used to describe the intellectual content of a manuscript or manuscript part. It comprises *either* a series of informal prose paragraphs *or* a series of <msItem> or <msItemStruct> elements, each of which provides a more detailed description of a single item contained within the manuscript. These may be prefaced, if desired, by a <summary> element, which is especially useful where one wishes to provide an overview of a manuscript's contents and describe only some of the items in detail.

<msContents> (manuscript contents) describes the intellectual content of a manuscript or manuscript part, either as a series of paragraphs or as a series of structured manuscript items.

<msItem> (manuscript item) describes an individual work or item within the intellectual content of a manuscript or manuscript part.

<msItemStruct> (structured manuscript item) contains a structured description for an individual work or item within the intellectual content of a manuscript or manuscript part.

In the simplest case, only a brief description may be provided, as in the following examples:

```
<msContents>
  <p>A collection of Lollard sermons</p>
</msContents>
<msContents>
  <p>Atlas of the world from Western Europe and Africa to Indochina,
     containing 27 maps and 26 tables</p>
</msContents>
<msContents>
  <p>Biblia sacra: Antiguo y Nuevo Testamento, con prefacios, prólogos
     y argumentos de san Jerónimo y de otros. Interpretaciones de los
     nombres hebreos.</p>
</msContents>
```

This description may of course be expanded to include any of the TEI elements generally available within a <p> element, such as <title>, <bibl>, or <list>. More usually, however, each individual work within a manuscript will be given its own description, using the <msItem> or <msItemStruct> element described in the next section, as in the following example:

```
<msContents>
  <msItem n="1">
    <locus>fols. 5r -7v</locus>
    <title>An ABC</title>
    <bibl>
      <title>IMEV</title>
      <biblScope>239</biblScope>
    </bibl>
  </msItem>
  <msItem n="2">
    <locus>fols. 7v -8v</locus>
    <title xml:lang="fr">Lenvoy de Chaucer a Scogan</title>
    <bibl>
      <title>IMEV</title>
      <biblScope>3747</biblScope>
    </bibl>
  </msItem>
  <msItem n="3">
    <locus>fol. 8v</locus>
    <title>Truth</title>
```

```
    <bibl>
      <title>IMEV</title>
      <biblScope>809</biblScope>
    </bibl>
  </msItem>
  <msItem  n="4">
    <locus>fols. 8v-10v</locus>
    <title>Birds Praise of Love</title>
    <bibl>
      <title>IMEV</title>
      <biblScope>1506</biblScope>
    </bibl>
  </msItem>
  <msItem  n="5">
    <locus>fols. 10v -11v</locus>
    <title  xml:lang="la">De amico ad amicam</title>
    <title  xml:lang="la">Responcio</title>
    <bibl>
      <title>IMEV</title>
      <biblScope>16 & 19</biblScope>
    </bibl>
  </msItem>
  <msItem  n="6">
    <locus>fols. 14r-126v</locus>
    <title>Troilus and Criseyde</title>
    <note>Bk. 1:71-Bk. 5:1701, with additional losses due to
        mutilation throughout</note>
  </msItem>
</msContents>
```

10.6.1 The <msItem> and <msItemStruct> Elements

Each discrete item in a manuscript or manuscript part can be described within a distinct <msItem> or <msItemStruct> element, and may be classified using the **class** attribute.

These are the possible component elements of <msItem> and <msItemStruct>.

<author> in a bibliographic reference, contains the name of the author(s), personal or corporate, of a work; the primary statement of responsibility for any bibliographic item.

<respStmt> (statement of responsibility) supplies a statement of responsibility for the intellectual content of a text, edition, recording, or series, where the specialized elements for authors, editors, etc. do not suffice or do not apply.

<title> contains the full title of a work of any kind.

@**type** classifies the title according to some convenient typology.

<rubric> contains the text of any rubric or heading attached to a particular manuscript item, that is, a string of words through which a manuscript signals the beginning of a text division, often with an assertion as to its author and title, which is in some way set off from the text itself, usually in red ink, or by use of different size or type of script, or some other such visual device.

<incipit> contains the incipit of a manuscript item, that is the opening words of the text proper, exclusive of any rubric which might precede it, of sufficient length to identify the work uniquely; such incipts were, in fomer times, frequently used a means of reference to a work, in place of a title.

<quote> (quotation) contains a phrase or passage attributed by the narrator or author to some agency external to the text.

<explicit> contains the explicit of a manuscript item, that is, the closing words of the text proper, exclusive of any rubric or colophon which might follow it.

<finalRubric> contains the string of words that denotes the end of a text division, often with an assertion as to its author and title, usually set off from the text itself by red ink, by a different size or type of script, or by some other such visual device.

<colophon> contains the colophon of a manuscript item: that is, a statement providing information regarding the date, place, agency, or reason for production of the manuscript.

<decoNote> (note on decoration) contains a note describing either a decorative component of a manuscript, or a fairly homogenous class of such components.

<listBibl> (citation list) contains a list of bibliographic citations of any kind.

<bibl> (bibliographic citation) contains a loosely-structured bibliographic citation of which the sub-components may or may not be explicitly tagged.

<filiation> contains information concerning the manuscript's filiation, i.e. its relationship to other surviving manuscripts of the same text, its protographs, antigraphs and apographs.

<note> contains a note or annotation.

@**type** describes the type of note.

<textLang> (text language) describes the languages and writing systems used by a manuscript (as opposed to its description, which is described in the <langUsage> element).

In addition, a <msItemStruct> may contain nested <msItemStruct> elements, just as an <msItem> may contain nested <msItem> elements.

The main difference between <msItem> and <msItemStruct> is that in the former, the order and number of child elements is not constrained; any element, in other words, may be given in any order, and repeated as often as is judged necessary. In the latter, however, the sub-elements, if used, must be given in the order specified above and only some of them may be repeated; specifically, <rubric>, <finalRubric>. <incipit>, <textLang> and <explicit> can appear only once.

While neither <msItem> nor <msItemStruct> may contain untagged running text, both permit an unstructured description to be provided in the form of one or more paragraphs of text. They differ in this respect also: if paragraphs are supplied as the content of an <msItem>, then none of the other component elements listed above is permitted; in the <msItemStruct> case, however, paragraphs may appear anywhere as an alternative to any of the component elements listed above.

As noted above, both <msItem> and <msItemStruct> elements may also nest, where a number of separate items in a manuscript are grouped under a single title or rubric, as is the case, for example, with a work like *The Canterbury Tales*.

The elements <msContents>, <msItem>, <msItemStruct>, <incipit>, and <explicit> are all members of the class att.msExcerpt from which they inherit the **defective** attribute.

att.msExcerpt (manuscript excerpt) provides attributes used to describe excerpts from a manuscript placed in a description thereof.

@**defective** indicates whether the passage being quoted is defective, i.e. incomplete through loss or damage.

This attribute can be used for example with collections of fragments, where each fragment is given as a separate <msItem> and the first and last words of each fragment are transcribed as defective incipits and explicits, as in the following example, a manuscript containing four fragments of a single work:

```
<msContents>
  <msItem  defective="true">
```

```
   <locus  from="1r"  to="9v">1r-9v</locus>
   <title>Knýtlinga saga</title>
   <msItem  n="1.1">
     <locus  from="1r:1"  to="2v:30">1r:1-2v:30</locus>
     <incipit  defective="true">dan<ex>n</ex>a a
           engl<ex>an</ex>di</incipit>
     <explicit  defective="true">en meðan <expan>haraldr</expan>
           hein hafði k<ex>onung</ex>r v<am>
        <g
           ref="http://www.examples.com/abbrevs.xml#er"/>
      </am>it
           yf<ex>ir</ex> danmork</explicit>
   </msItem>
<!-- msItems 1.2 to 1.4 -->
  </msItem>
</msContents>
```

The elements <ex>, <am>, and <expan> used in the above example are further discussed in section *11.3.2. Abbreviation and Expansion*; they are available only when the transcr module defined by that chapter is selected. Similarly, the <g> element used in this example to represent the abbreviation mark is defined by the gaiji module documented in chapter *5. Representation of Non-standard Characters and Glyphs*.

10.6.2 Authors and Titles

When used within a manuscript description, the <title> element should be used to supply a regularized form of the item's title, as distinct from any rubric quoted from the manuscript. If the item concerned has a standardized distinctive title, e.g. *Roman de la Rose*, then this should be the form given as content of the <title> element, with the value of the **type** attribute given as `uniform`. If no uniform title exists for an item, or none has been yet identified, or if one wishes to provide a general designation of the contents, then a 'supplied' title can be given, e.g. *missal*, in which case the **type** attribute on the <title> should be given the value `supplied`.

Similarly, if used within a manuscript description, the <author> element should always contain the normalized form of an author's name, irrespective of how (or whether) this form of the name is cited in the manuscript. If it is desired to retain the form of the author's name as given in the manuscript, this may be tagged as a distinct <name> element, within the text at the point where it occurs.

Note that the **key** attribute can also be used, as on names in general, to specify the identifier of a <person> element carrying full details of the person concerned (see further *10.3.6. Names of Persons, Places, and Organizations*).

The <respStmt> element can be used to supply the name and role of a person other than the author who is responsible for some aspect of the intellectual content of the manuscript:

```
<author>Diogenes Laertius</author>
<respStmt>
  <resp>in the translation of</resp>
  <name>Ambrogio Traversari</name>
</respStmt>
```

The <respStmt> element can also be used where there is a discrepancy between the author of an item as given in the manuscript and the accepted scholarly view, as in the following example:

```
<title type="supplied">Sermons on the Epistles and the Gospels</title>
<respStmt>
  <resp>here erroneously attributed to</resp>
  <name>St. Bonaventura</name>
</respStmt>
```

Note that such attributions of authorship, both correct and incorrect, are frequently found in the rubric or final rubric (and occasionally also elsewhere in the text), and can therefore be transcribed and included in the description, if desired, using the <rubric>, <finalRubric>, or <quote> elements, as appropriate.

10.6.3 Rubrics, Incipits, Explicits, and Other Quotations from the Text

It is customary in a manuscript description to record the opening and closing words of a text as well as any headings or colophons it might have, and the specialised elements <rubric>, <incipit>, <explicit>, <finalRubric>, and <colophon> are available within <msItem> for doing so, along with the more general <quote>, for recording other bits of the text not covered by these elements. Each of these elements has the same substructure, containing a mixture of phrase-level elements and plain text. A <locus> element can be included within each, in order to specify the location of the component, as in the following example:

```
<msContents>
  <msItem>
    <locus>f. 1-223</locus>
    <author>Radulphus Flaviacensis</author>
    <title>Expositio super Leviticum </title>
    <incipit>
      <locus>f. 1r</locus>
        Forte Hervei monachi</incipit>
    <explicit>
      <locus>f. 223v</locus>
        Benedictio salis et aquae</explicit>
  </msItem>
</msContents>
```

In the following example, standard TEI elements for the transcription of primary sources have been used to mark the expansion of abbreviations and other features present in the original:

```
<msItem defective="true">
  <locus>ff. 1r-24v</locus>
  <title type="uniform">Ágrip af Noregs konunga sǫgum</title>
  <incipit defective="true">regi oc h<ex>ann</ex> seti
    ho<gap reason="illegible" extent="7" unit="mm"/>
   <lb/>sc heim se<ex>m</ex> þio</incipit>
  <explicit defective="true">h<ex>on</ex> hev<ex>er</ex>
   <ex>oc</ex> þa buit hesta .ij. <lb/>annan viþ fé en
    h<ex>on</ex>o<ex>m</ex> annan til reiþ<ex>ar</ex>
  </explicit>
</msItem>
```

Note here also the use of the **defective** attribute on <incipit> and <explicit> to indicate that the text begins and ends defectively.

The **xml:lang** attribute for <colophon>, <explicit>, <incipit>, <quote>, and <rubric> may always be used to identify the language of the text quoted, if this is different from the default language specified by the **mainLang** attribute on <textLang>.

10.6.4 Filiation

The <filiation> element can be used to provide information on the relationship between the manuscript and other surviving manuscripts of the same text, either specifically or in a general way, as in the following example:

```
<msItem>
  <locus>118rb</locus>
  <incipit>Ecce morior cum nichil horum ... <ref>[Dn 13, 43]</ref>. Verba ista dixit Susanna de illis</incipit>
  <explicit>ut bonum comune conservatur.</explicit>
  <bibl>Schneyer 3, 436 (Johannes Contractus OFM)</bibl>
  <filiation>weitere Überl. Uppsala C 181, 35r.</filiation>
</msItem>
```

10.6.5 Text Classification

One or more text classification or text-type codes may be specified, either for the whole of the <msContents> element, or for one or more of its constituent <msItem> elements, using the **class** attribute as specified above:

```
<msContents>
  <msItem  n="1"  defective="false"  class="#law">
    <locus  from="1v"  to="71v">1v-71v</locus>
    <title  type="uniform">Jónsbók</title>
    <incipit>Magnus m<ex>ed</ex> guds miskun Noregs
        k<ex>onungu</ex>r</incipit>
    <explicit>en<ex>n</ex> u<ex>ir</ex>da
        þo t<ex>il</ex> fullra aura</explicit>
  </msItem>
</msContents>
```

The value of the **class** attribute should specify the identifier used for the appropriate classification within a <taxonomy> element, defined in the <classDecl> element of the TEI Header (*2.3.6. The Classification Declaration*), as shown here:

```
<classDecl>
  <taxonomy>
<!-- -->
    <category  xml:id="law">
      <catDesc>Laws</catDesc>
    </category>
<!-- -->
  </taxonomy>
</classDecl>
```

10.6.6 Languages and Writing Systems

The <textLang> element should be used to provide information about the languages used within a manuscript item. It may take the form of a simple note, as in the following example:

```
<textLang>Old Church Slavonic, written in Cyrillic script.</textLang>
```

Where, for validation and indexing purposes, it is thought convenient to add keywords identifying the particular languages used, the **mainLang** attribute may be used. This attribute takes the same range of values as the global **xml:lang** attribute, on which see further *vi.1 Language identification.* In the following example a manuscript written chiefly in Old Church Slavonic is described:

```
<textLang mainLang="chu">Old Church Slavonic</textLang>
```

A manuscript item will sometimes contain material in more than one language. The **mainLang** attribute should be used only for the chief language. Other languages used may be specified using the **otherLangs** attribute as in the following example:

```
<textLang mainLang="chu" otherLangs="RUS HEL">Mostly Old Church
Slavonic, with some Russian and Greek material</textLang>
```

Since Old Church Slavonic may be written in either Cyrillic or Glagolitic scripts, and even occasionally in both within the same manuscript, it might be preferable to use a more explicit identifier:

```
<textLang mainLang="chu-Cyrs">Old Church Slavonic in Cyrillic script</textLang>
```

The form and scope of language identifiers recommended by these Guidelines is based on the IANA standard described at *vi.1 Language identification* and should be followed throughout. Where additional detail is needed correctly to describe a language, or to discuss its deployment in a given text, this should be done using the <langUsage> element in the TEI Header, within which individual <language> elements document the languages used: see *2.4.2. Language Usage*.

Note that the <language> element defines a particular combination of human language and writing system. Only one <language> element may be supplied for each such combination. Standard TEI practice also allows this element to be referenced by any element using the global **xml:lang** attribute in order to specify the language applicable to the content of that element. For example, assuming that <language> elements have been defined with the identifiers fr (for French), la (for Latin), and de (for German), a manuscript description written in French which specifies that a particular manuscript contains predominantly German but also some Latin material, might have a <textLang> element like the following:

```
<textLang xml:lang="fr" mainLang="de" otherLangs="la">allemand et latin</textLang>
```

10.7 Physical Description

Under the general heading 'physical description' we subsume a large number of different aspects generally regarded as useful in the description of a given manuscript. These include:

- aspects of the form, support, extent, and quire structure of the manuscript object and of the way in which the text is laid out on the page (*10.7.1. Object Description*);
- the styles of writing, such as the way it is laid out on the page, the styles of writing, decorative features, any musical notation employed and any annotations or marginalia (*10.7.2. Writing, Decoration, and Other Notations*);
- and discussion of its binding, seals, and any accompanying material (*10.7.3. Bindings, Seals, and Additional Material*).

Most manuscript descriptions touch on several of these categories of information though few include them all, and not all distinguish them as clearly as we propose here. In particular, it is often the case that an existing description will include information for which we propose distinct elements within a single paragraph, or even sentence. The encoder must then decide whether to rewrite the description using the structure proposed here, or to retain the existing prose, marked up simply as a series of <p> elements, directly within the <physDesc> element.

The <physDesc> element may thus be used in either of two distinct ways. It may contain a series of paragraphs addressing topics listed above and similar ones. Alternatively, it may act as a container for any choice of the more specialized elements described in the remainder of this section, each of which itself contains a series of paragraphs, and may also have more specific attributes. Note that the two ways must *not* be combined within the same description.

10.7.1 Object Description

The <objectDesc> element is used to group together those parts of the physical description which relate specifically to the text-bearing object, its format, constitution, layout, etc. The **form** attribute is used to indicate the specific type of writing vehicle being described, for example, as a codex, roll, tablet, etc. The <objectDesc> element has two parts: a description of the support, i.e. the physical carrier on which the text is inscribed; and a description of the layout, i.e. the way text is organized on the carrier.

Taking these in turn, the description of the support is tagged using the following elements, each of which is discussed in more detail below:

<supportDesc> (support description) groups elements describing the physical support for the written part of a manuscript.

<support> contains a description of the materials etc. which make up the physical support for the written part of a manuscript.

<extent> describes the approximate size of a text as stored on some carrier medium, whether digital or non-digital, specified in any convenient units.

<collation> contains a description of how the leaves or bifolia are physically arranged.

<foliation> describes the numbering system or systems used to count the leaves or pages in a codex.

<condition> contains a description of the physical condition of the manuscript.

Each of these elements contains paragraphs relating to the topic concerned. Within these paragraphs, phrase-level elements (in particular those discussed above at *10.3. Phrase-level Elements*), may be used to tag specific terms of interest if so desired.

```
<objectDesc  form="codex">
  <supportDesc>
    <p>Mostly <material>paper</material>, with watermarks
    <watermark>unicorn</watermark> (<ref>Briquet 9993</ref>) and
    <watermark>ox</watermark> (close to <ref>Briquet 2785</ref>).
        The first and last leaf of each quire, with the exception of
        quires xvi and xviii, are constituted by bifolia of parchment,
        and all seven miniatures have been painted on inserted
        singletons of parchment.</p>
  </supportDesc>
</objectDesc>
```

This example combines information which might alternatively be more precisely tagged using the more specific elements described in the following subsections.

10.7.1.1 Support

The <support> element groups together information about the physical carrier. Typically, for western manuscripts, this will entail discussion of the material (parchment, paper, or a combination of the two) written on. For paper, a discussion of any watermarks present may also be useful. If this discussion makes reference to standard catalogues of such items, these may be tagged using the standard <ref> element as in the following example:

```
<support>
  <p>
    <material>Paper</material> with watermark: <watermark>anchor in a circle
        with star on top</watermark>, <watermark>countermark B-B with
        trefoil</watermark> similar to <ref>Moschin, Anchor N 1680</ref>
    <date>1570-1585</date>.</p>
</support>
```

10.7.1.2 Extent

The <extent> element, defined in the TEI header, may also be used in a manuscript description to specify the number of leaves a manuscript contains, as in the following example:

```
<extent>ii + 97 + ii</extent>
```

Information regarding the size of the leaves may be specifically marked using the phrase level <dimensions> element, as in the following example, or left as plain prose.

```
<extent>ii + 321 leaves
<dimensions  unit="cm">
    <height>35</height>
    <width>27</width>
  </dimensions>
</extent>
```

Alternatively, the generic <measure> element might be used within <extent>, as in the following example:

```
<extent>
  <measure  type="composition"  unit="leaf"  quantity="10">10 Bl.</measure>
  <measure  type="height"  quantity="37"  unit="cm">37</measure> x
<measure  type="width"  quantity="29"  unit="cm">29</measure> cm

</extent>
```

10.7.1.3 Collation

The <collation> element should be used to provide a description of a book's current and original structure, that is, the arrangement of its leaves and quires. This information may be conveyed using informal prose, or any appropriate notational convention. Although no specific notation is defined here, an appropriate element to enclose such an expression would be the <formula> element, which is provided when the figures module is included in a schema. Here are some examples of different ways of treating collation:

```
<collation>
  <p>
    <formula>1-3:8, 4:6, 5-13:8</formula>
  </p>
</collation>
<collation>
  <p>There are now four gatherings, the first, second and fourth originally consisting of
     eight leaves, the third of seven. A fifth gathering thought to have followed has left no trace.
  <list>
```

```
     <item>Gathering I consists of 7 leaves, a first leaf, originally conjoint with <locus>fol. 7</locus>,
           having been cut away leaving only a narrow strip along the gutter; the others, <locus>fols 1</locus>

             and <locus>6</locus>, <locus>2</locus> and <locus>5</locus>, and <locus>3</locus> and
<locus>4</locus>,
           are bifolia.</item>
     <item>Gathering II consists of 8 leaves, 4 bifolia.</item>
     <item>Gathering III consists of 7 leaves; <locus>fols 16</locus> and <locus>22</locus> are conjoint,
           the others singletons.</item>
     <item>Gathering IV consists of 2 leaves, a bifolium.</item>
   </list>
  </p>
</collation>
<collation>
  <p>I (1, 2+9, 3+8, 4+7, 5+6, 10); II (11, 12+17, 13, 14, 15, 16, 18, 19).</p>
</collation>
<collation>
  <p>
    <formula>1-5.8 6.6 (catchword, f. 46, does not match following
        text) 7-8.8 9.10, 11.2 (through f. 82) 12-14.8 15.8(-7)</formula>
  </p>
</collation>
```

10.7.1.4 Foliation

The <foliation> element may be used to indicate the scheme, medium or location of folio, page, column, or line numbers written in the manuscript, frequently including a statement about when and, if known, by whom, the numbering was done.

```
<foliation>
  <p>Neuere Foliierung, die auch das Vorsatzblatt mitgezählt hat.</p>
</foliation>
<foliation>
  <p>Folio numbers were added in brown ink by Árni Magnússon
     ca. 1720-1730 in the upper right corner of all recto-pages.</p>
</foliation>
```

Where a manuscript contains traces of more than one foliation, each should be recorded as a distinct <foliation> element and optionally given a distinct value for its **xml:id** attribute. The <locus> element discussed in *10.3.5. References to Locations within a Manuscript* can then indicate which foliation scheme is being cited by means of its **scheme** attribute, which points to this identifier:

```
<foliation  xml:id="original">
  <p>Original foliation in red roman numerals in the middle of
     the outer margin of each recto</p>
</foliation>
<foliation  xml:id="modern">
  <p>Foliated in pencil in the top right
     corner of each recto page.</p>
</foliation>
<!-- ... -->
<locus  scheme="#modern">ff 1-20</locus>
```

10.7.1.5 Condition

The <condition> element is used to summarize the overall physical state of a manuscript, in particular where such information is not recorded elsewhere in the description. It should not, however, be used to describe changes or repairs to a manuscript, as these are more appropriately described as a part of its custodial history (see *10.7.5.1.2. Availability and Custodial History*).

```
<condition>
  <p>The manuscript shows signs of damage from water and mould on its outermost leaves.</p>
</condition>
<condition>
  <p>Despite tears on many of the leaves the codex is reasonably well preserved.
     The top and the bottom of f. 1 is damaged, and only a thin slip is left of the original second
     leaf (now foliated as 1bis). The lower margin of f. 92 has been cut away. There is a lacuna of
     one leaf between ff. 193 and 194. The manuscript ends defectively (there are approximately six
     leaves missing).</p>
</condition>
```

10.7.1.6 Layout Description

The second part of the <objectDesc> element is the <layoutDesc> element, which is used to describe and document the *mise-en-page* of the manuscript, that is the way in which text and illumination are arranged on the page, specifying for example the number of written, ruled, or pricked lines and columns per page, size of margins, distinct blocks such as glosses, commentaries, etc. This may be given as a simple series of paragraphs. Alternatively, one or more different layouts may be identified within a single manuscript, each described by its own <layout> element.

<layoutDesc> (layout description) collects the set of layout descriptions applicable to a manuscript.

<layout> describes how text is laid out on the page, including information about any ruling, pricking, or other evidence of page-preparation techniques.

Where the <layout> element is used, the layout will often be sufficiently regular for the attributes on this element to convey all that is necessary; more usually however a more detailed treatment will be required. The attributes are provided as a convenient shorthand for commonly occurring cases, and should not be used except where the layout is regular. The value `NA` (not-applicable) should be used for cases where the layout is either very irregular, or where it cannot be characterized simply in terms of lines and columns, for example, where blocks of commentary and text are arranged in a regular but complex pattern on each page

The following examples indicate the range of possibilities:

```
<layout  ruledLines="25 32">
  <p>Most pages have between 25 and 32 long lines ruled in lead.</p>
</layout>
<layout  columns="1"  writtenLines="24">
  <p>Written in one column throughout; 24 lines per page.</p>
</layout>
<layout>
  <p>Written in 3 columns, with 8 lines of text and interlinear glosses in
     the centre, and up to 26 lines of gloss in the outer two columns. Double
     vertical bounding lines ruled in hard point on hair side. Text lines ruled
     faintly in lead. Remains of prickings in upper, lower, and outer (for 8 lines
     of text only) margins.</p>
</layout>
```

Where multiple <layout> elements are supplied, the scope for each specification can be indicated by means of <locus> elements within the content of the element, as in the following example:

```
<layoutDesc>
  <layout  ruledLines="25 32">
    <p>On <locus  from="1r"  to="202v">fols 1r-200v</locus> and
    <locus  from="210r"  to="212v">fols 210r-212v</locus> there are
        between 25 and 32 ruled lines.</p>
  </layout>
  <layout  ruledLines="34 50">
    <p>On <locus  from="203r"  to="209v">fols 203r-209v</locus> there are between 34
        and 50 ruled lines.</p>
  </layout>
</layoutDesc>
```

10.7.2 Writing, Decoration, and Other Notations

The second group of elements within a structured physical description concerns aspects of the writing, illumination, or other notation (notably, music) found in a manuscript, including additions made in later hands — the 'text', as it were, as opposed to the carrier.

<handDesc> (description of hands) contains a description of all the different kinds of writing used in a manuscript.

<handNote> (note on hand) describes a particular style or hand distinguished within a manuscript.

<decoDesc> (decoration description) contains a description of the decoration of a manuscript, either as a sequence of paragraphs, or as a sequence of topically organised <decoNote> elements.

<decoNote> (note on decoration) contains a note describing either a decorative component of a manuscript, or a fairly homogenous class of such components.

<musicNotation> contains description of type of musical notation.

<additions> contains a description of any significant additions found within a manuscript, such as marginalia or other annotations.

10.7.2.1 Writing

The <handDesc> element can contain a short description of the general characteristics of the writing observed in a manuscript, as in the following example:

```
<handDesc>
  <p>Written in a <term>late Caroline minuscule</term>; versals in a
     form of <term>rustic capitals</term>; although the marginal and
     interlinear gloss is written in varying shades of ink that are
     not those of the main text, text and gloss appear to have been
     copied during approximately the same time span.</p>
</handDesc>
```

Note the use of the <term> element to mark specific technical terms within the context of the <handDesc> element.

Where several distinct hands have been identified, this fact can be registered by using the **hands** attribute, as in the following example:

```
<handDesc  hands="2">
  <p>The manuscript is written in two contemporary hands, otherwise
     unknown, but clearly those of practised scribes. Hand I writes
     ff. 1r-22v and hand II ff. 23 and 24. Some scholars, notably
```

```
    Verner Dahlerup and Hreinn Benediktsson, have argued for a third hand
    on f. 24, but the evidence for this is insubstantial.</p>
</handDesc>
```

Alternatively, or in addition, where more specific information about one or more of the hands identified is to be recorded, the <handNote> element should be used, as in the following example:

```
<handDesc hands="3">
  <handNote xml:id="Eirsp-1" scope="minor">
    <p>The first part of the manuscript,
    <locus from="1v" to="72v:4">fols 1v-72v:4</locus>, is written in a practised
        Icelandic Gothic bookhand. This hand is not found elsewhere.</p>
  </handNote>
  <handNote xml:id="Eirsp-2" scope="major">
    <p>The second part of the manuscript, <locus from="72v:4" to="194v">fols
           72v:4-194</locus>, is written in a hand contemporary with the first; it can
        also be found in a fragment of <title>Knýtlinga saga</title>,
    <ref>AM 20b II fol.</ref>.</p>
  </handNote>
  <handNote xml:id="Eirsp-3" scope="minor">
    <p>The third hand has written the majority of the chapter headings.
        This hand has been identified as the one also found in <ref>AM
           221 fol.</ref>.</p>
  </handNote>
</handDesc>
```

Note here the use of the <locus> element, discussed in section *10.3.5. References to Locations within a Manuscript*, to specify exactly which parts of a manuscript are written by a given hand.

When a full or partial transcription of a manuscript is available in addition to the manuscript description, the <handShift> element described in *11.4.1. Document Hands* can be used to link the relevant parts of the transcription to the appropriate <handNote> element in the description: for example, at the point in the transcript where the second hand listed above starts (i.e. at folio 72v:4), we might insert **<handShift new="#Eirsp-2"/>**.

10.7.2.2 Decoration

It can be difficult to draw a clear distinction between aspects of a manuscript which are purely physical and those which form part of its intellectual content. This is particularly true of illuminations and other forms of decoration in a manuscript. We propose the following elements for the purpose of delimiting discussion of these aspects within a manuscript description, and for convenience locate them all within the physical description, despite the fact that the illustrative features of a manuscript will in many cases also be seen as constitutiing part of its intellectual content.

The <decoDesc> element may contain simply one or more paragraphs summarizing the overall nature of the decorative features of the manuscript, as in the following example:

```
<decoDesc>
  <p>The decoration comprises two full page miniatures, perhaps added
     by the original owner, or slightly later; the original major decoration
     consists of twenty-three large miniatures, illustrating the divisions of
     the Passion narrative and the start of the major texts, and the major
     divisions of the Hours; seventeen smaller miniatures, illustrating the
     suffrages to saints; and seven historiated initials, illustrating
```

```
    the pericopes and major prayers.</p>
</decoDesc>
```

Alternatively, it may contain a series of more specific typed <decoNote> elements, each summarizing a particular aspect or individual instance of the decoration present, for example the use of miniatures, initials (historiated or otherwise), borders, diagrams, etc., as in the following example:

```
<decoDesc>
  <decoNote type="miniature">
    <p>One full-page miniature, facing the beginning of the first
        Penitential Psalm.</p>
  </decoNote>
  <decoNote type="initial">
    <p>One seven-line historiated initial, commencing the first
        Penitential Psalm.</p>
  </decoNote>
  <decoNote type="initial">
    <p>Six four-line decorated initials, commencing the second through the
        seventh Penitential Psalm.</p>
  </decoNote>
  <decoNote type="initial">
    <p>Some three hundred two-line versal initials with pen-flourishes,
        commencing the psalm verses.</p>
  </decoNote>
  <decoNote type="border">
    <p>Four-sided border decoration surrounding the miniatures and three-sided
        border decoration accompanying the historiated and decorated initials.</p>
  </decoNote>
</decoDesc>
```

Where more exact indexing of the decorative content of a manuscript is required, the standard TEI elements <term> or <index> may be used within the prose description to supply or delimit appropriate iconographic terms, as in the following example:

```
<decoDesc>
  <decoNote type="miniatures">
    <p>Fourteen large miniatures with arched tops, above five lines of text:
    <list>
       <item>
         <locus>fol. 14r</locus>Pericopes. <term>St. John writing on
                  Patmos</term>, with the Eagle holding his ink-pot and pen-case; some
              flaking of pigment, especially in the sky</item>
       <item>
         <locus>fol. 26r</locus>Hours of the Virgin, Matins.
       <term>Annunciation</term>; Gabriel and the Dove to the right</item>
       <item>
         <locus>fol. 60r</locus>Prime. <term>Nativity</term>; the
       <term>Virgin and Joseph adoring the Child</term>
       </item>
       <item>
         <locus>fol. 66r</locus>Terce. <term>Annunciation to the
                  Shepherds</term>, one with <term>bagpipes</term>
       </item>
<!-- ... -->
```

```
    </list>
   </p>
  </decoNote>
</decoDesc>
```

10.7.2.3 Musical Notation

Where a manuscript contains music, the <musicNotation> element may be used to describe the form of notation employed, as in the following example:

```
<musicNotation>
  <p>Square notation on 4-line red staves.</p>
</musicNotation>
<musicNotation>
  <p>Neumes in campo aperto of the St. Gall type.</p>
</musicNotation>
```

10.7.2.4 Additions and Marginalia

The <additions> element can be used to list or describe any additions to the manuscript, such as marginalia, scribblings, doodles, etc., which are considered to be of interest or importance. Such topics may also be discussed or referenced elsewhere in a description, for example in the <history> element, in cases where the marginalia provide evidence of ownership. Some examples follow:

```
<additions>
  <p>Doodles on most leaves, possibly by children, and often quite amusing.</p>
</additions>
<additions>
  <p  xml:lang="fr">Quelques annotations marginales des XVIe et XVIIe s.</p>
</additions>
<additions>
  <p>The text of this manuscript is not interpolated with sentences from
     Royal decrees promulgated in 1294, 1305 and 1314. In the margins, however,
     another somewhat later scribe has added the relevant paragraphs of these
     decrees, see pp. 8, 24, 44, 47 etc.</p>
  <p>As a humorous gesture the scribe in one opening of the manuscript, pp. 36
     and 37, has prolonged the lower stems of one letter f and five letters þ
     and has them drizzle down the margin.</p>
</additions>
<additions>
  <p>Spaces for initials and chapter headings were left by the scribe but not filled in.
     A later, probably fifteenth-century, hand has added initials and chapter headings in
     greenish-coloured ink on fols <locus>8r</locus>, <locus>8v</locus>, <locus>9r</locus>,
  <locus>10r</locus> and <locus>11r</locus>. Although a few of these chapter headings are
     now rather difficult to read, most can be made out, e.g. fol. <locus>8rb</locus>
    <quote  xml:lang="is">floti ast<ex>ri</ex>d<ex>ar</ex>
    </quote>; fol. <locus>9rb</locus>
    <quote  xml:lang="is">v<ex>m</ex> olaf conung</quote>, and fol. <locus>10ra</locus>
    <quote  xml:lang="is">Gipti<ex>n</ex>g ol<ex>a</ex>fs k<ex>onun</ex>gs</quote>.</p>
  <p>The manuscript contains the following marginalia:
  <list>
     <item>Fol. <locus>4v</locus>, left margin: <quote  xml:lang="is">hialmadr <ex>ok</ex>
        <lb/>brynjadr</quote>,
           in a fifteenth-cenury hand, imitating an addition made to the text by the scribe at this point.</item>
```

```
     <item>Fol. <locus>5r</locus>, lower margin: <quote  xml:lang="is">þ<ex>e</ex>tta þiki
               m<ex>er</ex> v<ex>er</ex>a gott blek en<ex>n</ex>da kan<ex>n</ex> ek icki
                betr sia</quote>, in a fifteenth-century hand, probably the same as that on the previous
page.</item>
     <item>Fol. <locus>9v</locus>, bottom margin: <quote  xml:lang="is">þessa bok uilda eg <sic>gæt</sic>
               lært med <lb/>an Gud gefe myer Gott ad <lb/>læra</quote>; seventeenth-century hand.</item>
   </list>
  </p>
  <p>There are in addition a number of illegible scribbles in a later hand (or hands) on fols
  <locus>2r</locus>, <locus>3r</locus>, <locus>5v</locus> and <locus>19r</locus>.</p>
</additions>
```

10.7.3 Bindings, Seals, and Additional Material

The third major component of the physical description relates to supporting but distinct physical components, such as bindings, seals and accompanying material. These may be described using the following specialist elements:

<bindingDesc> (binding description) describes the present and former bindings of a manuscript, either as a series of paragraphs or as a series of distinct <binding> elements, one for each binding of the manuscript.

<binding> contains a description of one binding, i.e. type of covering, boards, etc. applied to a manuscript.

<sealDesc> (seal description) describes the seals or other external items attached to a manuscript, either as a series of paragraphs or as a series of distinct <seal> elements, possibly with additional <decoNote>s.

<seal> contains a description of one seal or similar attachment applied to a manuscript.

<accMat> (accompanying material) contains details of any significant additional material which may be closely associated with the manuscript being described, such as non-contemporaneous documents or fragments bound in with the manuscript at some earlier historical period.

10.7.3.1 Binding Descriptions

The <bindingDesc> element contains a description of the state of the present and former bindings of a manuscript, including information about its material, any distinctive marks, and provenance information. This may be given as a series of paragraphs, if only one binding is being described, or as a series of distinct <binding> elements, each describing a distinct binding, where these are separately described. For example:

```
<bindingDesc>
  <p>Sewing not visible; tightly rebound over 19th-century pasteboards, reusing
    panels of 16th-century brown leather with gilt tooling à la fanfare, Paris
    c. 1580-90, the centre of each cover inlaid with a 17th-century oval medallion
    of red morocco tooled in gilt (perhaps replacing the identifying mark of a
    previous owner); the spine similarly tooled, without raised bands or title-piece;
    coloured endbands; the edges of the leaves and boards gilt. Boxed.</p>
</bindingDesc>
```

Within a binding description, the element <decoNote> is available, as an alternative to <p>, for paragraphs dealing exclusively with information about decorative features of a binding, as in the following example:

```
<binding>
  <p>Bound, s. XVIII (?), in <material>diced russia leather</material>
     retaining most of the original 15th century metal ornaments (but with
     some replacements) as well as the heavy wooden boards.</p>
  <decoNote>
    <p>On each cover: alternating circular stamps of the Holy Monogram,
        a sunburst, and a flower.</p>
  </decoNote>
  <decoNote>
    <p>On the cornerpieces, one of which is missing, a rectangular stamp
        of the Agnus Dei.</p>
  </decoNote>
  <p>Rebacked during the 19th century.</p>
</binding>
```

10.7.3.2 Seals

The <sealDesc> element supplies information about the seal(s) attached to documents to guarantee their integrity, or to show authentication of the issuer or consent of the participants. It may contain one or more paragraphs summarizing the overall nature of the seals, or may contain one or more <seal> elements.

```
<sealDesc>
  <seal  n="1"  type="pendant"  subtype="cauda_duplex">
    <p>Round seal of <name>Anders Olufsen</name> in black wax:
    <bibl>
       <ref>DAS 930</ref>
     </bibl>. Parchment tag, on which is written:
    <quote>pertinere nos predictorum placiti nostri iusticarii precessorum dif</quote>.</p>
  </seal>
  <seal  n="2"  type="pendant"  subtype="cauda_duplex">
    <p>The seal of <name>Jens Olufsen</name> in black wax:
    <bibl>
       <ref>DAS 1061</ref>
     </bibl>. Legend: <quote>S IOHANNES OLAVI</quote>.
        Parchment tag on which is written: <quote>Woldorp Iohanne G</quote>.</p>
  </seal>
</sealDesc>
```

10.7.3.3 Accompanying Material

The circumstance may arise where material not originally part of a manuscript is bound into or otherwise kept with a manuscript. In some cases this material would best be treated in a separate <msPart> element (see *10.7.6. Manuscript Parts* below). There are, however, cases where the additional matter is not self-evidently a distinct manuscript: it might, for example, be a set of notes by a later scholar, or a file of correspondence relating to the manuscript. The <accMat> element is provided as a holder for this kind of information.

<accMat> (accompanying material) contains details of any significant additional material which may be closely associated with the manuscript being described, such as non-contemporaneous documents or fragments bound in with the manuscript at some earlier historical period.

Here is an example of the use of this element, describing a note by the Icelandic manuscript collector Árni Magnússon which has been bound with the manuscript:

```
<accMat>
  <p>A slip in Árni Magnússon's hand has been stuck to the
     pastedown on the inside front cover; the text reads:
  <quote  xml:lang="is">Þidreks Søgu þessa hefi eg
        feiged af Sekreterer Wielandt Anno 1715 i Kaupmanna høfn. Hun er,
        sem eg sie, Copia af Austfirda bókinni (Eidagás) en<ex>n</ex>
        ecki progenies Brædratungu bokarinnar. Og er þar fyrer eigi i
        allan<ex>n</ex> máta samhlioda þ<ex>eir</ex>re er
        Sr Jon Erlendz son hefer ritad fyrer Mag. Bryniolf. Þesse Þidreks
        Saga mun vera komin fra Sr Vigfuse á Helgafelle.</quote>
  </p>
</accMat>
```

10.7.4 History

The following elements are used to record information about the history of a manuscript:

<history> groups elements describing the full history of a manuscript or manuscript part.

<origin> contains any descriptive or other information concerning the origin of a manuscript or manuscript part.

<provenance> contains any descriptive or other information concerning a single identifiable episode during the history of a manuscript or manuscript part, after its creation but before its acquisition.

<acquisition> contains any descriptive or other information concerning the process by which a manuscript or manuscript part entered the holding institution.

The three components of the <history> element all have the same substructure, consisting of one or more paragraphs marked as <p> elements. Each of these three elements is also a member of the att.datable attribute class, itself a member of the att.datable.w3c class, and thus also carries the following optional attributes:

att.datable.w3c provides attributes for normalization of elements that contain datable events using the W3C datatypes.

@**notBefore** specifies the earliest possible date for the event in standard form, e.g. yyyy-mm-dd.

@**notAfter** specifies the latest possible date for the event in standard form, e.g. yyyy-mm-dd.

Information about the origins of the manuscript, its place and date of writing, should be given as one or more paragraphs contained by a single <origin> element; following this, any available information on distinct stages in the history of the manuscript before its acquisition by its current holding institution should be included as paragraphs within one or more <provenance> elements. Finally, any information specific to the means by which the manuscript was acquired by its present owners should be given as paragraphs within the <acquisition> element.

Here is a fairly simple example of the use of this element:

```
<history>
  <origin>
    <p>Written in <origPlace>Durham</origPlace> during <origDate  notBefore="1125"  notAfter="1175">the
          mid-twelfth century</origDate>.</p>
  </origin>
  <provenance>
    <p>Recorded in two medieval catalogues of the books belonging
       to <name  type="org">Durham Priory</name>, made in <date>1391</date> and
    <date>1405</date>.</p>
    <p>Given to <name  type="person">W. Olleyf</name> by <name  type="person">William
```

```
          Ebchester, Prior (1446-56)</name> and later belonged to <name  type="person">Henry
          Dalton</name>, Prior of Holy Island (<name  type="place">Lindisfarne</name>)
       according to inscriptions on ff. 4v and 5.</p>
  </provenance>
  <acquisition>
    <p>Presented to <name  type="org">Trinity College</name> in
    <date>1738</date> by <name  type="person">Thomas Gale</name> and
       his son <name  type="person">Roger</name>.</p>
  </acquisition>
</history>
```

Here is a fuller example:

```
<history>
  <origin  notBefore="1225"  notAfter="1275">
    <p>Written in Spain or Portugal in the middle of the 13th century
       (the date 1042, given in a marginal note on f. 97v, cannot be correct.)</p>
  </origin>
  <provenance>
    <p>The Spanish scholar <name  type="person">Benito Arias
          Montano</name> (1527-1598) has written his name on f. 97r, and may be
       presumed to have owned the manuscript. It came somehow into the
       possession of <foreign  xml:lang="da">etatsråd</foreign>
     <name  type="person">Holger Parsberg</name> (1636-1692), who has written his
       name twice, once on the front pastedown and once on f. 1r, the former dated
    <date>1680</date> and the latter <date>1682</date>. Following Parsberg's
       death the manuscript was bought by <foreign>etatsråd</foreign>
     <name  type="person">Jens Rosenkrantz</name> (1640-1695) when Parsberg's
       library was auctioned off (23 October 1693).</p>
  </provenance>
  <acquisition  notBefore="1696"  notAfter="1697">
    <p>The manuscript was acquired by Árni
       Magnússon from the estate of Jens Rosenkrantz, presumably at
       auction (the auction lot number 468 is written in red chalk on the
       flyleaf), either in 1696 or 97.</p>
  </acquisition>
</history>
```

10.7.5 Additional information

Three categories of additional information are provided for by the scheme described here, grouped together within the <additional> element described in this section.

<additional> groups additional information, combining bibliographic information about a manuscript, or surrogate copies of it with curatorial or administrative information.

<adminInfo> (administrative information) contains information about the present custody and availability of the manuscript, and also about the record description itself.

<surrogates> contains information about any digital or photographic representations of the manuscript being described which may exist in the holding institution or elsewhere.

<listBibl> (citation list) contains a list of bibliographic citations of any kind.

None of the constituent elements of <additional> is required. If any is supplied, it may appear once only; furthermore, the order in which elements are supplied should be as specified above.

10.7.5.1 Administrative information

The <adminInfo> element is used to hold information relating to the curation and management of a manuscript. This may be supplied as a note using the global <note> element. Alternatively, different aspects of this information may be presented grouped within one of the following specialized elements:

<recordHist> (recorded history) provides information about the source and revision status of the parent manuscript description itself.

<availability> supplies information about the availability of a text, for example any restrictions on its use or distribution, its copyright status, etc.

<custodialHist> (custodial history) contains a description of a manuscript's custodial history, either as running prose or as a series of dated custodial events.

10.7.5.1.1 Record History

The <recordHist> element may contain simply a series of paragraphs. Alternatively it may contain a <source> element, followed by an optional series of <change> elements.

<source> describes the original source for the information contained with a manuscript description.

<change> summarizes a particular change or correction made to a particular version of an electronic text which is shared between several researchers.

The <source> element is used to document the primary source of information for the record containing it, in a similar way to the standard TEI <sourceDesc> element within a TEI Header. If the record is a new one, made without reference to anything other than the manuscript itself, then it may simply contain a <p> element, as in the following example:

```
<source>
  <p>Directly catalogued from the original manuscript.</p>
</source>
```

Frequently, however, the record will be derived from some previously existing description, which may be specified using the standard TEI <bibl> element, as in the following example:

```
<recordHist>
  <source>
    <p>Information transcribed from <bibl>
       <title>The index of Middle English verse</title>
       <biblScope>123</biblScope>
     </bibl>.</p>
  </source>
</recordHist>
```

If, as is likely, a full bibliographic description of the source from which cataloguing information was taken is included within the <listBibl> element contained by the current <additional> element, or elsewhere in the current document, then it need not be repeated here. Instead, it should be referenced using the standard TEI <ref> element, as in the following example:

```
<additional>
  <adminInfo>
    <recordHist>
      <source>
        <p>Information transcribed from
```

```
      <bibl>
         <ref  target="#IMEV">IMEV</ref> 123</bibl>.</p>
     </source>
   </recordHist>
  </adminInfo>
  <listBibl>
    <bibl  xml:id="IMEV">
     <author>Carleton Brown</author> and <author>Rossell Hope Robbins</author>
     <title  level="m">The index of Middle English verse</title>
     <pubPlace>New York</pubPlace>
     <date>1943</date>
    </bibl>
<!-- other bibliographic records relating to this manuscript here -->
  </listBibl>
</additional>
```

The <change> element may also appear within the <revisionDesc> element of the standard TEI Header; its use here is intended to signal the similarity of function between the two container elements. Where the TEI Header should be used to document the revision history of the whole electronic file to which it is prefixed, the <recordHist> element may be used to document changes at a lower level, relating to the individual description, as in the following example:

```
<change  when="2005-03-10">On 10 March 2005
<name>MJD</name> added provenance information</change>
```

10.7.5.1.2 Availability and Custodial History

The <availability> element is another element also available in the TEI Header, which should be used here to supply any information concerning access to the current manuscript, such as its physical location (where this is not implicit in its identifier), any restrictions on access, information about copyright, etc.

```
<availability>
  <p>Viewed by appointment only, to be arranged with curator.</p>
</availability>
<availability>
  <p>In conservation, Jan.  - Mar., 2002.  On loan to the
     Bayerische Staatsbibliothek, April - July, 2002.</p>
</availability>
<availability>
  <p>The manuscript is in poor condition, due to many of the leaves being
     brittle and fragile and the poor quality of a number of earlier repairs;
     it should therefore not be used or lent out until it has been conserved.</p>
</availability>
```

The <custodialHist> record is used to describe the custodial history of a manuscript, recording any significant events noted during the period that it has been located within its holding institution. It may contain either a series of <p> elements, or a series of <custEvent> elements, each describing a distinct incident or event, further specified by a **type** attribute, and carrying dating information by virtue of its membership in the att.datable class, as noted above.

<custEvent> (custodial event) describes a single event during the custodial history of a manuscript.

Here is an example of the use of this element:

```
<custodialHist>
  <custEvent type="conservation" notBefore="1961-03-01" notAfter="1963-02-28">
    <p>Conserved between March 1961 and February 1963 at Birgitte Dalls
        Konserveringsværksted.</p>
  </custEvent>
  <custEvent type="photography" notBefore="1988-05-01" notAfter="1988-05-30">
    <p>Photographed in May 1988 by AMI/FA.</p>
  </custEvent>
  <custEvent type="transfer" notBefore="1989-11-13" notAfter="1989-11-13">
    <p>Dispatched to Iceland 13 November 1989.</p>
  </custEvent>
</custodialHist>
```

10.7.5.2 Surrogates

The <surrogates> element is used to provide information about any digital or photographic representations of the manuscript which may exist within the holding institution or elsewhere.

<surrogates> contains information about any digital or photographic representations of the manuscript being described which may exist in the holding institution or elsewhere.

The <surrogates> element should not be used to repeat information about representations of the manuscript available within published works; this should normally be documented within the <listBibl> element within the <additional> element. However, it is often also convenient to record information such as negative numbers or digital identifiers for unpublished collections of manuscript images maintained within the holding institution, as well as to provide more detailed descriptive information about the surrogate itself. Such information may be provided as prose paragraphs, within which identifying information about particular surrogates may be presented using the standard TEI <bibl> element, as in the following example:

```
<surrogates>
  <p>
    <bibl>
      <title type="gmd">microfilm (master)</title>
      <idno>G.neg. 160</idno> n.d.</bibl>
    <bibl>
      <title type="gmd">microfilm (archive)</title>
      <idno>G.pos. 186</idno> n.d.</bibl>
    <bibl>
      <title type="gmd">b/w prints</title>
      <idno>AM 795 4to</idno>
      <date when="1999-01-27">27 January 1999</date>
      <note>copy of G.pos. 186</note>
    </bibl>
    <bibl>
      <title type="gmd">b/w prints</title>
      <idno>reg.nr. 75</idno>
      <date when="1999-01-25">25 January 1999</date>
      <note>photographs of the spine, outside covers, stitching etc.</note>
    </bibl>
  </p>
</surrogates>
```

Note the use of the specialized form of title (general material designation) to specify the kind of surrogate being documented.

At a later revision, the content of the <surrogates> element is likely to be expanded to include elements more specifically intended to provide detailed information such as technical details of the process by which a digital or photographic image was made. For information about the inclusion of digital facsimile images within a TEI document, refer also to *11.1. Digital Facsimiles*.

10.7.6 Manuscript Parts

The <msPart> element may be used in cases where what were originally physically separate manuscripts or parts of manuscripts have been bound together and/or share the same call number.

<msPart> (manuscript part) contains information about an originally distinct manuscript or part of a manuscript, now forming part of a composite manuscript.

Since each component of such a composite manuscript will in all likelihood have its own content, physical description, history, and so on, the structure of <msPart> is in the main identical to that of <msDesc>, allowing one to retain the top level of identity (<msIdentifier>), but to branch out thereafter into as many parts, or even subparts, as necessary. If the parts of a composite manuscript have their own identifiers, they should be tagged using the <idno> element, rather than the <msIdentifier> element, as in the following example:

```
<msDesc>
  <msIdentifier>
    <settlement>Amiens</settlement>
    <repository>Bibliothèque Municipale</repository>
    <idno>MS 3</idno>
    <msName>Maurdramnus Bible</msName>
  </msIdentifier>
<!-- other elements here -->
  <msPart>
    <altIdentifier>
      <idno>MS 6</idno>
    </altIdentifier>
<!-- other information specific to this part here -->
  </msPart>
  <msPart>
    <altIdentifier>
      <idno>MS 7</idno>
    </altIdentifier>
<!-- other information specific to this part here -->
  </msPart>
  <msPart>
    <altIdentifier>
      <idno>MS 9</idno>
    </altIdentifier>
<!-- other information specific to this part here -->
  </msPart>
<!-- other msParts here -->
</msDesc>
```

10.7.7 Module for Manuscription Description

The module described in this chapter makes available the following components:

Module msdescription: Manuscript Description

- Elements defined: accMat acquisition additional additions adminInfo altIdentifier binding bindingDesc catchwords collation collection colophon condition custEvent custodialHist decoDesc decoNote depth dimensions explicit filiation finalRubric foliation handDesc height heraldry history

incipit institution layout layoutDesc locus material msContents msDesc msIdentifier msItem msItemStruct msName msPart musicNotation objectDesc origDate origPlace origin physDesc provenance recordHist repository rubric seal sealDesc secFol signatures source stamp summary support supportDesc surrogates textLang watermark width

- Classes defined: att.msExcerpt

The selection and combination of modules to form a TEI schema is described in *1.2. Defining a TEI Schema*.

Chapter 11

Representation of Primary Sources

This chapter defines a module intended for use in the representation of primary sources, such as manuscripts or other written materials. Section *11.1. Digital Facsimiles* provides elements for the encoding of digital facsimiles or images of such materials, while the remainder of the chapter discusses ways of encoding detailed transcriptions of such materials. It is expected that this module will also be useful in the preparation of critical editions, but the module defined here is distinct from that defined in chapter *12. Critical Apparatus*, and may be used independently of it. Detailed metadata relating to primary sources of any kind may be recorded using the elements defined by the manuscript description module discussed in chapter *10. Manuscript Description*, but again the present module may be used independently if such data is not required.

It should be noted that, as elsewhere in these Guidelines, this chapter places more emphasis on the problems of representing the textual components of a document than on those relating to the description of the document's physical characteristics such as the carrier medium or physical construction. These aspects, of particular importance in codicology and the bibliographic study of incunables, are touched on in the chapter on Manuscript Description (*10. Manuscript Description*) and also form the subject of ongoing work in the TEI Physical Bibliography workgroup.

Although this chapter discusses manuscript materials more frequently than other forms of written text, most of the recommendations presented are equally applicable *mutatis mutandis* in the encoding of printed matter or indeed any form of written source, including monumental inscriptions. Similarly, where in the following descriptions terms such as 'scribe', 'author', 'editor', 'annotator' or 'corrector' are used, these may be re-interpreted in terms more appropriate to the medium being transcribed. In printed material, for example, the 'compositor' plays a role analogous to the 'scribe', while in an authorial manuscript, the author and the scribe are the same person.

11.1 Digital Facsimiles

These Guidelines are mostly concerned with the preparation of digital texts, in which a pre-existing text is transcribed or otherwise converted into character form, and marked up in XML. However, it is also very common practice to make a different form of 'digital text', which is instead composed of digital images of the original source, typically one per page, or other written surface. We call such a resource a digital facsimile. A digital facsimile may, in the simplest case, just consist of a collection of images, with some metadata to identify them and the source materials portrayed. It may sometimes contain a variety of images of the same source pages, for example of different resolutions, or of different kinds. Such a collection may form part of any kind of document, for example a commentary of a codicological or paeleographic nature, where there is a need to align explanatory text with image data. And it may also be complemented by a transcribed or encoded version of the original source, which may be linked to the page images. In this section we present elements designed to support these various possibilities and discuss the associated mechanisms provided by these Guidelines.

When this module is included in a schema, the class att.global is extended to include a new pointer attribute **facs**:

att.global.facs groups elements corresponding with all or part of an image, because they contain an alternative representation of it, typically but not necessarily a transcription of it.

@**facs** (facsimile) points to all or part of an image which corresponds with the content of the element.

This attribute may be used to associate any element in a transcribed text with an image of it, by means of the usual URI pointing mechanism.

If a digital text contains one image per page or column (or similar unit), and no more complex mapping between text and image is envisaged, then the **facs** attribute may be used to point directly to a graphic resource:

```
<TEI>
  <teiHeader>
<!--...-->
  </teiHeader>
  <text>
    <pb  facs="page1.png"/>
<!-- text contained on page 1 is encoded here -->
    <pb  facs="page2.png"/>
<!-- text contained on page 2 is encoded here -->
  </text>
</TEI>
```

By convention, this encoding indicates that the image indicated by **facs** attribute represents the whole of the text following the <pb> (pagebreak) element, up to the next <pb> element. Any convenient milestone element (see further *3.10.3. Milestone Elements*) could be used in the same way; for example if the images represent individual columns, the <cb> element might be used. Though simple, this method has some drawbacks. It does not scale well to more complex cases where, for example, the images do not correspond exactly with transcribed pages, or where the intention is to align specific marked up elements with detailed images, or parts of images. And it makes the management of the information about the images more difficult by scattering references to them through the file. Nevertheless, this solution may be adequate for many straightforward 'digital library' applications.

The recommended approach to encoding facsimiles is instead to use the **facs** attribute in conjunction with the elements <facsimile>, <surface>, and <zone>, which are also provided by this module. These elements make it possible to accommodate multiple images of each page, as well as to record arbitrary planar coordinates of textual elements on any kind of written surface and to link such elements with digital facsimile images of them. Typical applications include the provision of full text search in 'digital facsimile editions', and ways of annotating graphics, for example so as to identify individuals appearing in a group portraits and link them to data about the person represented.

The following elements are used to represent components of a digital facsimile:

<facsimile> contains a representation of some written source in the form of a set of images rather than as transcribed or encoded text.

<surface> defines a written surface in terms of a rectangular coordinate space, optionally grouping one or more graphic representations of that space, and rectangular zones of interest within it.

@**start** points to an element which encodes the starting position of the text corresponding to the inscribed part of the surface.

<zone> defines a rectangular area contained within a <surface> element.

The <facsimile> element is used to represent a digital facsimile. It appears within a TEI document along with, or instead of, the <text> element introduced in section *4. Default Text Structure*. When this module is selected therefore, a legal TEI document may thus comprise any of the following:-

- a TEI Header and a text element
- a TEI Header and a facsimile element
- a TEI Header, a facsimile element, and a text element

Like the <text> element, a <facsimile> element may also contain an optional <front> or <back> element, used in the same way as described in sections *4.5. Front Matter* and *4.7. Back Matter*.

In the simplest case, a facsimile just contains a series of <graphic> elements, each of which identifies an image file:

```
<facsimile>
  <graphic url="page1.png"/>
  <graphic url="page2.png"/>
  <graphic url="page3.png"/>
  <graphic url="page4.png"/>
</facsimile>
```

If desired, the <binaryObject> element described in *3.9. Graphics and other non-textual components* (or any other element from the model.graphicLike class) can be used instead of a <graphic>.

In this simple case, the four page images are understood to represent the complete facsimile, and are to be read in the sequence given. Suppose, however, that the second page of this particular work is available both as an ordinary photograph and as an infra-red image, or in two different resolutions. The <surface> element may be used to indicate that there are two image files corresponding with the same area of the work:

```
<facsimile>
  <graphic url="page1.png"/>
  <surface>
    <graphic url="page2-highRes.png"/>
    <graphic url="page2-lowRes.png"/>
  </surface>
  <graphic url="page3.png"/>
  <graphic url="page4.png"/>
</facsimile>
```

The <surface> element provides a way of indicating that the two images of page2 represent the same physical surface within the source material. A surface might be a sheet of paper or parchment, a face of a monument, a billboard, a membrane of a scroll, or indeed any two-dimensional surface, of any size.

The actual dimensions of the object represented are not documented by the <surface> element; instead, the <surface> is located within an abstract coordinate space, which is defined by the following attributes, supplied by the att.coordinated class:

att.coordinated elements which can be positioned within a two dimensional coordinate system.

- @**ulx** gives the x coordinate value for the upper left corner of a rectangular space.
- @**uly** gives the y coordinate value for the upper left corner of a rectangular space.
- @**lrx** gives the x coordinate value for the lower right corner of a rectangular space.
- @**lry** gives the y coordinate value for the lower right corner of a rectangular space.

The same coordinate space is used for a <surface> and for all of its child elements.[1] It may be most convenient to derive a coordinate space from a digital image of the surface in question such that each pixel in the image corresponds with a whole number of units (typically 1) in the coordinate space. In other cases it may be more convenient to use units such as millimetres; in neither case is any specific mapping to the physical dimensions of the object represented implied.

Each <surface> can contain one or more <zone> elements, each of which represents a rectangular region or bounding box defined in terms of the same coordinate space as that of its parent <surface> element. This provides a unit of analysis which may be used to define any rectangular region of interest, such as a detail or illustration, or some part of the surface which is to be aligned with a particular text element. The att.coordinated attributes listed above are also used to supply the coordinates of a zone.

As we have seen, a surface will usually correspond with the whole of a written surface. A zone, by contrast, defines any arbitrary rectangular area of interest using the same coordinate system. It might be bigger or smaller than its parent surface, or might overlap its boundaries. The only constraint is that it must be defined using the same coordinate system.

When an image of some kind is supplied within either a zone or a surface, the implication is that the whole of the image represents the zone or surface containing it. In the simple case therefore, we might imagine a surface defining a page, within which there is a graphic representing the whole of that page, and a number of zones defining parts of the page, each with its own graphic, each representing a part of the page. If however one of those graphics actually represents an area larger than the page (for example to include a binding or the surface of a desk on which the page rests), then it will be enclosed by a zone with coordinates larger than those of the parent surface.

Note that this mechanism does not provide any way of addressing a non-rectangular area, nor of coping with distortions introduced by perspective or parallax; if this is needed, the more powerful mechanisms provided by the Standard Vector Graphics (SVG) language should be used to define an overlay, as further discussed in *16.4.3. A Three-way Alignment*.

For example, consider the following figure: This is an image of a two page spread from a manuscript in the



Figure 11.1: Relation between page, surface, and zone

Badische Landesbibliothek, Karlsruhe. We have no information as to the dimensions of the original object, but the low resolution image displayed here contains 500 pixels horizontally and 321 pixels vertically. For convenience, we might map each pixel to one cell of the coordinate space.[2]

The coordinates of the <surface> (that is, the area of the image which represents the written two page spread) can then be specified in terms of this coordinate space, simply by counting pixels in the image. The left corner of the two page spread appears 50 units from the left of the image and 20 units from the top, while the

[1] The coordinate space may be thought of as a grid superimposed on a rectangular space. Rectangular areas of the grid are defined as four numbers **a b c d**: the first two identify the grid point which is at the upper left corner of the rectangle; the second two give the grid point located at the lower right corner of the rectangle. The grid point **a b** is understood to be the point which is located **a** points from the origin along the **x** (horizontal) axis, and **b** points from the origin along the **y** (vertical) axis.

[2] The coordinate space used here is based on pixels, but the mapping between pixels and units in the coordinate space need not be one-to-one; it might be convenient to define a more delicate grid, to enable us to address much smaller parts of the image. This can be done simply by supplying appropriate values for the attributes which define the coordinate space; for example doubling them all would map each pixel to two grid points in the coordinate space.

bottom right corner of the spread appears 400 units from the left of the image, and 280 units from the top. We therefore define the written surface within this image as follows:

```
<facsimile>
  <surface
     ulx="50"
     uly="20"
     lrx="400"
     lry="280">
<!-- ... -->
  </surface>
</facsimile>
```

To describe the whole image, we will also need to define a zone of interest which represents an area larger than this surface. Using the same coordinate system as that defined for the surface, its coordinates are **0,0,500,321**. This zone of interest can be defined by a <zone> element, within which we can place the uncropped <graphic>:

```
<facsimile>
  <surface
     ulx="50"
     uly="20"
     lrx="400"
     lry="280">
    <zone
       ulx="0"
       uly="0"
       lrx="500"
       lry="321">
      <graphic
         url="http://upload.wikimedia.org/wikipedia/commons/5/50/Handschrift.karlsruhe.blb.jpg"/>
    </zone>
  </surface>
</facsimile>
```

If desired, the <binaryObject> element described in *3.9. Graphics and other non-textual components* (or any other element from the model.graphicLike class) may be used instead of a <graphic> element.

The <desc> element may also be used within either <surface> or <zone> to provide some further information about the area being defined. For example, since the image in this example contains two pages, it might be preferable to define two distinct surfaces, one for each page, including its illuminated margins. In this case, each surface must specify a bounding box which encloses the appropriate page, as well as defining the zone for the graphic itself:

```
<facsimile>
  <surface
     ulx="50"
     uly="20"
     lrx="210"
     lry="280">
    <desc>left hand page</desc>
    <zone
       ulx="0"
       uly="0"
       lrx="500"
```

```
      lry="321">
     <graphic
        url="http://upload.wikimedia.org/wikipedia/commons/5/50/Handschrift.karlsruhe.blb.jpg"/>
   </zone>
  </surface>
  <surface
     ulx="240"
     uly="25"
     lrx="400"
     lry="280">
   <desc>right hand page</desc>
   <zone
      ulx="0"
      uly="0"
      lrx="500"
      lry="321">
     <graphic
        url="http://upload.wikimedia.org/wikipedia/commons/5/50/Handschrift.karlsruhe.blb.jpg"/>
   </zone>
  </surface>
</facsimile>
```

In addition to acting as a container for <graphic> elements, <zone> elements may also be used to select parts of each surface for analytical purposes. For example, to define the written part of the left hand page:

```
<facsimile>
  <surface
     ulx="50"
     uly="20"
     lrx="210"
     lry="280">
   <desc>Left hand page</desc>
   <zone
      ulx="0"
      uly="0"
      lrx="500"
      lry="321">
     <graphic
        url="http://upload.wikimedia.org/wikipedia/commons/5/50/Handschrift.karlsruhe.blb.jpg"/>
   </zone>
   <zone
      ulx="90"
      uly="40"
      lrx="200"
      lry="225">
     <desc>Written part of left hand page</desc>
   </zone>
  </surface>
</facsimile>
```

In the following example, we discuss a hypothetical digital edition of an early 16th century French work, Charles de Bovelles' *Géometrie Pratique*.[3] In this edition, each page has been digitized as a separate file: for example, recto page 49 is stored in a file called Bovelles-49r.png. In the <facsimile> element used to contain

[3] The image is taken from the collection at `http://ancilla.unice.fr/Illustr.html`, and was digitized from a copy in the Bibliothèque Municipale de Lyon, by whose kind permission it is included here

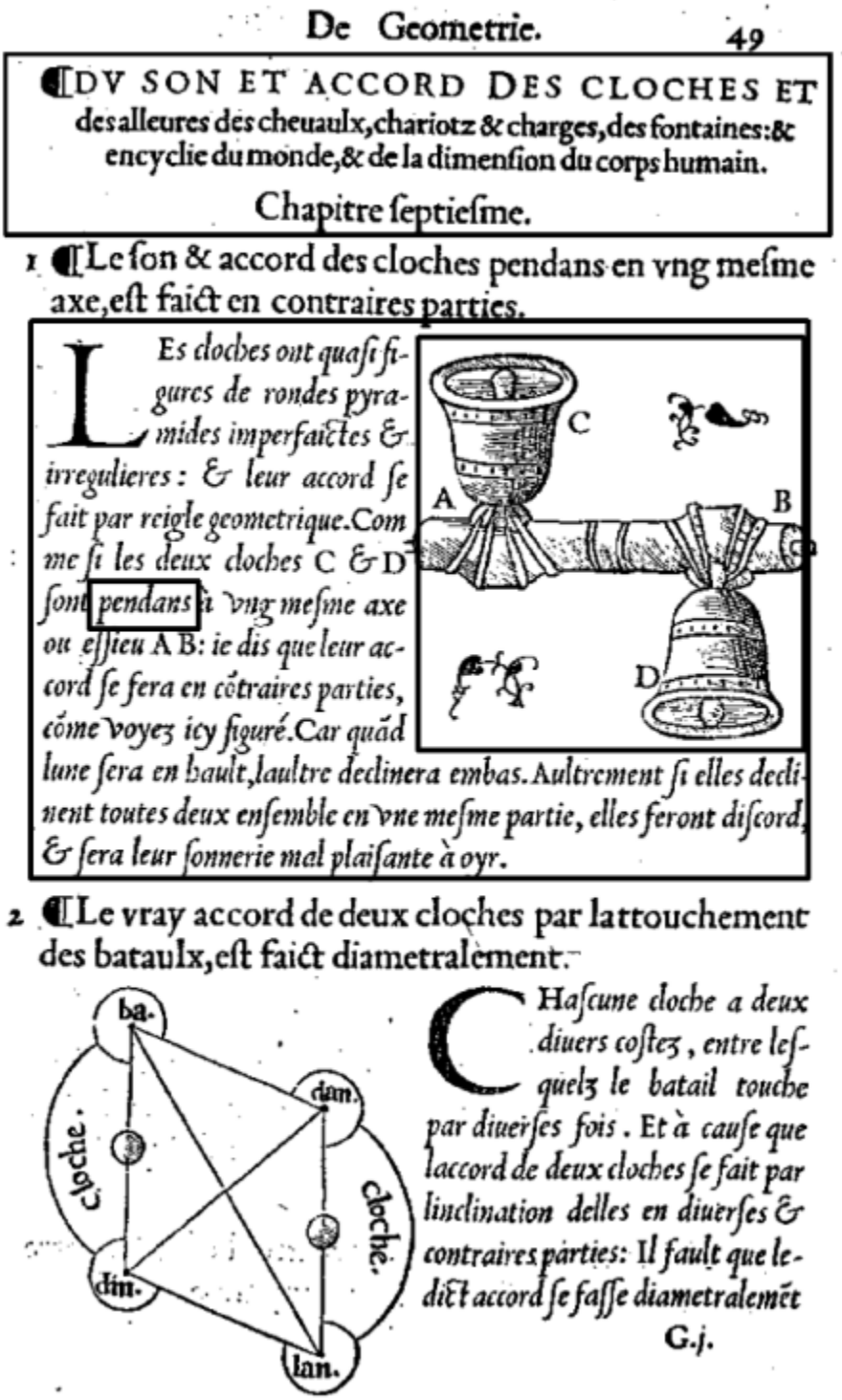
the whole set of pages, we define a <surface> element for this page, which we situate within a coordinate scale running from 0 to 200 in the x (horizontal) axis, and 0 to 300 in the y (vertical) axis. The <surface> element contains a <graphic> element which represents the whole of this surface:

```
<facsimile>
  <surface
     ulx="0"
     uly="0"
     lrx="200"
     lry="300">
    <graphic  url="Bovelles-49r.png"/>
  </surface>
</facsimile>
```

We can now identify distinct zones within the page image using the coordinate scale defined for the surface. In Figure 3, Zones within a surface we show the upper part of the page, with boxes indicating four such zones. Each of these will be represented by a <zone> element, given within the <surface> element already defined, and specified in terms of the same coordinate system.

The following encoding defines each of the four zones identified in the figure.

```
<facsimile>
  <surface
     ulx="0"
     uly="0"
     lrx="200"
     lry="300">
    <graphic  url="Bovelles-49r.png"/>
    <zone
       ulx="25"
       uly="25"
       lrx="180"
       lry="60">
      <desc>contains the title</desc>
    </zone>
    <zone
       ulx="28"
       uly="75"
       lrx="175"
       lry="178"/>
<!-- contains the paragraph in italics -->
    <zone
       ulx="105"
       uly="76"
       lrx="175"
       lry="160"/>
<!-- contains the figure -->
    <zone
       ulx="45"
       uly="125"
       lrx="60"
       lry="130"/>
<!-- contains the word "pendans" -->
  </surface>
</facsimile>
```



De Geometrie. 49

DV SON ET ACCORD DES CLOCHES ET des alleures des cheuaulx, chariotz & charges, des fontaines: & encyclie du monde, & de la dimenſion du corps humain.

Chapitre ſeptieſme.

1 Le ſon & accord des cloches pendans en vng meſme axe, eſt faict en contraires parties.

Les cloches ont quaſi figures de rondes pyramides imperfaictes & irregulieres: & leur accord ſe fait par reigle geometrique. Comme ſi les deux cloches C & D ſont pendans à vng meſme axe ou eſſieu A B: ie dis que leur accord ſe fera en cõtraires parties, cõme voyez icy figuré. Car quãd lune ſera en hault, laultre declinera embas. Aultrement ſi elles declinent toutes deux enſemble en vne meſme partie, elles feront diſcord, & ſera leur ſonnerie mal plaiſante à oyr.

C
A
B
D

2 Le vray accord de deux cloches par lattouchement des bataulx, eſt faict diametralement.

ba.
dan.
cloche.
cloche.
din.
lan.

CHaſcune cloche a deux diuers coſtez, entre leſquelz le batail touche par diuerſes fois. Et à cauſe que laccord de deux cloches ſe fait par linclination delles en diuerſes & contraires parties: Il fault que ledict accord ſe faſſe diametralemẽt

G.j.

Figure 11.2: Zones within a surface

Note that the location of each zone is defined independently but using the same coordinate system, so that they may overlap freely. Zones need not nest within each other; they must however be rectangular, as previously noted. As noted earlier, a zone may fall outside the area of the surface which defines its coordinate space.

In this example a single <graphic> element has been associated directly with the surface of the page rather than nesting it within a zone. However, it is also possible to include multiple <zone> elements which contain a <graphic> element, if for example a detailed image is available. Since all <zone> elements use the same coordinate system (that defined by their parent <surface>), there is no need to demonstrate enclosure of one zone within another by means of nesting. To continue the current example, supposing that we have an additional image called Bovelles49r-detail.png containing an additional image of the figure in the third zone above, we might encode that zone as follows:

```
<zone
   ulx="105"
   uly="76"
   lrx="175"
   lry="160">
  <graphic  url="Bovelles49r-detail.png"/>
</zone>
```

Now suppose that we wish to align a transcription of this page with the zones identified above. The first step is to give each relevant part of the facsimile an identifier:

```
<facsimile>
  <surface
     ulx="0"
     uly="0"
     lrx="200"
     lry="300">
   <zone
      xml:id="B49r"
      ulx="0"
      uly="0"
      lrx="200"
      lry="300">
     <graphic  url="Bovelles-49r.png"/>
   </zone>
   <zone
      ulx="105"
      uly="76"
      lrx="175"
      lry="160">
     <graphic  url="Bovelles49r-detail.png"/>
   </zone>
   <zone
      xml:id="B49rHead"
      ulx="25"
      uly="25"
      lrx="180"
      lry="60"/>
<!-- contains the title -->
   <zone
      xml:id="B49rPara2"
      ulx="28"
      uly="75"
```

```
      lrx="175"
      lry="178"/>
<!-- contains the paragraph in italics -->
   <zone
      xml:id="B49rFig1"
      ulx="105"
      uly="76"
      lrx="175"
      lry="160"/>
<!-- contains the figure -->
   <zone
      xml:id="B49rW457"
      ulx="45"
      uly="125"
      lrx="60"
      lry="130"/>
<!-- contains the word "pendans" -->
  </surface>
</facsimile>
```

The alignment between transcription and image is made, as usual, by means of the **facs** attribute:

```
<pb  facs="#B49r"/>
<fw>De Geometrie 49</fw>
<head  facs="#B49rHead">DU SON ET ACCORD DES CLOCHES ET <lb/> des alleures des chevaulx,
chariotz & charges, des fontaines:& <lb/> encyclie du monde,
& de la dimension du corps humain.</head>
<head>Chapitre septiesme</head>
<div  n="1">
  <p>Le son & accord des cloches pendans en ung mesme <lb/> axe, est
     faict en contraires parties.</p>
  <p  rend="it"  facs="#B49rPara2">LEs cloches ont quasi fi<lb/>gures de rondes
     pyra<lb/>mides imperfaictes & <lb/> irregulieres: & leur
     accord se <lb/> fait par reigle geometrique. Com<lb/>me si les deux
     cloches C & D <lb/> sont <w  facs="#B49rW457">pendans</w> à ung
     mesme axe <lb/> ou essieu A B: je dis que leur ac<lb/>cord se fera en
     co<ex>n</ex>traires parties<lb/> co<ex>m</ex>me voyez icy
     figuré. Car qua<ex>n</ex>d <lb/> lune sera en hault, laultre
     declinera embas. Aultrement si elles decli<lb/>nent toutes deux
     ensembles en une mesme partie, elles seront discord, <lb/> & sera
     leur sonnerie mal plaisante à oyr.<figure  facs="#B49rFig1">
      <graphic  url="Bovelles49r-detail.png"/>
    </figure>
  </p>
</div>
```

Further discussion of the encoding choices made in the above transcription is provided in the remainder of this chapter.

It is also possible to point in the other direction, from a <surface> or <zone> to the corresponding text. This is the function of the **start** attribute, which supplies the identifier of the element containing the transcribed text found within the surface or zone concerned. Thus, another way of linking this page with its transcription would be simply

```
<facsimile>
  <surface start="#PB49R">
    <graphic url="Bovelles-49r.png"/>
  </surface>
</facsimile>
<text>
<!-- ... -->
  <pb xml:id="PB49R"/>
  <fw>De Geometrie 49</fw>
<!-- ... -->
</text>
```

11.2 Scope of Transcriptions

When transcribing a primary source, scholars may wish to record information concerning individual readings of letters, words, or larger units, whether the object is simply a 'neutral' transcription or a critical edition. In either case they may also wish to include other editorial material, such as comments on the status or possible origin of particular readings, corrections, or text supplied to fill lacunae. Further, it is customary in transcriptions to register certain features of the source, such as ornamentation, underlining, deletion, areas of damage and lacunae. This chapter provides ways of encoding such information:

- first, methods of recording editorial or other alterations to the text, such as expansion of abbreviations, corrections, conjectures, etc. (section *11.3. Altered, Corrected, and Erroneous Texts*)
- then, methods of describing important extra-linguistic phenomena in the source: unusual spaces, lines, page and line breaks, change of manuscript hand, etc. (section *11.4. Hands and Responsibility*)
- finally, a method of recording material such as running heads, catch-words, and the like (section *11.7. Headers, Footers, and Similar Matter*)

These recommendations are not intended to meet every transcriptional circumstance likely to be faced by any scholar. Rather, they should be regarded as a base which can be elaborated if necessary by different scholars in different disciplines.

As a rule, all elements which may be used in the course of a transcription of a single witness may also be used in a critical apparatus, i.e. within the elements proposed in chapter *12. Critical Apparatus*. This can generally be achieved by nesting a particular reading containing tagged elements from a particular witness within the <rdg> element in an <app> structure.

Just as a critical apparatus may contain transcriptional elements within its record of variant readings in various witnesses, one may record variant readings in an individual witness by use of the apparatus mechanisms <app> and <rdg>. This is discussed in section *12.3. Using Apparatus Elements in Transcriptions*.

11.3 Altered, Corrected, and Erroneous Texts

In the detailed transcription of any source, it may prove necessary to record various types of actual or potential alteration of the text: expansion of abbreviations, correction of the text (either by author, scribe, or later hand, or by previous or current editors or scholars), addition, deletion, or substitution of material, and the like. The sections below describe how such phenomena may be encoded using either elements defined in the core module (defined in chapter *3. Elements Available in All TEI Documents*) or specialized elements available only when the module described in this chapter is available.

11.3.1 Core elements for Transcriptional Work

In transcribing individual sources of any type, encoders may record corrections, normalizations, expansions of abbreviations, additions, and omissions using the elements described in section *3.4. Simple Editorial Changes*. Those particularly relevant to this chapter include:

<abbr> (abbreviation) contains an abbreviation of any sort.

<add> (addition) contains letters, words, or phrases inserted in the text by an author, scribe, annotator, or corrector.

<choice> groups a number of alternative encodings for the same point in a text.

<corr> (correction) contains the correct form of a passage apparently erroneous in the copy text.

<del> (deletion) contains a letter, word, or passage deleted, marked as deleted, or otherwise indicated as superfluous or spurious in the copy text by an author, scribe, annotator, or corrector.

<expan> (expansion) contains the expansion of an abbreviation.

<gap> indicates a point where material has been omitted in a transcription, whether for editorial reasons described in the TEI header, as part of sampling practice, or because the material is illegible, invisible, or inaudible.

<sic> (latin for thus or so) contains text reproduced although apparently incorrect or inaccurate.

Several of these elements bear additional attributes for specifying who is responsible for the interpretation represented by the markup, and the certainty associated with it. In addition, some of them bear an attribute allowing the markup to be categorised by type and source.

att.editLike provides attributes describing the nature of a encoded scholarly intervention or interpretation of any kind.

- @**cert** (certainty) signifies the degree of certainty associated with the intervention or interpretation.
- @**resp** (responsible party) indicates the agency responsible for the intervention or interpretation, for example an editor or transcriber.
- @**source** contains a list of one or more pointers indicating the sources which support the given reading.

att.typed provides attributes which can be used to classify or subclassify elements in any way.

- @**type** characterizes the element in some sense, using any convenient classification scheme or typology.
- @**subtype** provides a sub-categorization of the element, if needed

The specific aspect of the markup described by these attributes differs on different elements; for further discussion, see the relevant sections below, especially section *11.4.2. Hand, Responsibility, and Certainty Attributes*.

The following sections describe how the core elements just named may be used in the transcription of primary source materials.

11.3.2 Abbreviation and Expansion

The writing of manuscripts by hand lends itself to the use of abbreviation to shorten scribal labour. Commonly occurring letters, groups of letters, words, or even whole phrases, may be represented by significant marks. This phenomenon of manuscript abbreviation is so widespread and so various that no taxonomy of it is here attempted. Instead, methods are shown which allow abbreviations to be encoded using the core elements mentioned above.

A manuscript abbreviation may be viewed in two ways. One may transcribe it as a particular sequence of letters or marks upon the page: thus, a 'p with a bar through the descender', a 'superscript hook', a 'macron'.

One may also interpret the abbreviation in terms of the letter or letters it is seen as standing for: thus, 'per', 're', 'n'. Both of these views are supported by these Guidelines.

In many cases the glyph found in the manuscript source also exists in the Unicode character set: for example the common Latin brevigraph ⁊, standing for *et* and often known as the 'Tironian et' can be directly represented in any XML document as the Unicode character with code point **U+204A** (see further *v.6.1 Character References* and *vi.1 Language identification*). In cases where it does not, these Guidelines recommend use of the <g> element provided by the gaiji module described in chapter *5. Representation of Non-standard Characters and Glyphs*. This module allows the encoder great flexibility both in processing and in documenting non-standard characters or glyphs, including the ability to provide detailed documentation and images for them.

These two methods of coding abbreviation may also be combined. An encoder may record, for any abbreviation, both the sequence of letters or marks which constitutes it, and its sense, that is, the letter or letters for which it is believed to stand. For example, in the following fragment the phrase *euery persone* is represented by a sequence of characters which may be transcribed directly, using the <g> element to indicate the two brevigraphs it contains as follows:

```
eu<g ref="#er">er</g>y <g ref="#per">per</g>sone that loketh after heuen hath a place in this
ladder
```

Source: [137]

Note that in each case the <g> element may contain a suggested replacement for the referenced brevigraph; this is purely advisory however, and may not be appropriate in all cases.

The transcriber may also wish to indicate that, because of the presence of these particular characters, the two words are actually abbreviations, by using the <abbr> element:

```
<abbr>eu<g ref="#er">er</g>y</abbr>
<abbr>
  <g ref="#per">per</g>sone
</abbr>
...
```

Alternatively, the transcriber may choose silently to expand these abbreviations, using the <expan> element:

```
<expan>euery</expan>
<expan>persone</expan> ...
```

And, of course, the <choice> element can be used to show that one encoding is an alternative for the other:

```
<choice>
  <abbr>eu<g ref="#er">er</g>y</abbr>
  <expan>euery</expan>
</choice>
```

When abbreviated forms such as these are expanded, two processes are carried out: some characters not present in the abbreviation are added (always), and some characters or glyphs present in the abbreviation are omitted or replaced (often). For example, when the abbreviation *Dr.* is expanded to *Doctor*, the dot in the abbreviation is removed, and the letters *octo* are added. Where detailed markup of abbreviated words is required, these two aspects may be marked up explicitly, using the following elements:

<ex> (editorial expansion) contains a sequence of letters added by an editor or transcriber when expanding an abbreviation.

<am> (abbreviation marker) contains a sequence of letters or signs present in an abbreviation which are omitted or replaced in the expanded form of the abbreviation.

Using these elements, a transcriber may indicate the status of the individual letters or signs within both the abbreviation and the expansion. The <am> element surrounds characters or signs such as tittles or tildes, used to indicate the presence of an abbreviation, which are typically removed or replaced by other characters in the expanded form of the abbreviation:

```
<abbr>eu<am>
   <g ref="#er"/>
  </am>y</abbr>
<abbr>
  <am>
    <g ref="#per"/>
  </am>sone
</abbr> ...
```

while the <ex> element may be used to indicate those characters within the expansion which are not present in the abbreviated form.

```
<expan>eu<ex>er</ex>y</expan>
<expan>
  <ex>per</ex>sone
</expan> ...
```

The content of the <abbr> element should usually include the whole of the abbreviated word, while the <expan> element should include the whole of its expansion. If this is not considered necessary, the <am> and <ex> elements may be used within a <choice> element, as in this example:

```
eu<choice>
  <am>
    <g ref="#er"/>
  </am>
  <ex>er</ex>
</choice>y
<choice>
  <am>
    <g ref="#per"/>
  </am>
  <ex>per</ex>
</choice>sone ...
```

As implied in the preceding discussion, making decisions about which of these various methods of representing abbreviation to use will form an important part of an encoder's practice. As a rule, the <abbr> and <am> elements should be preferred where it is wished to signify that the content of the element is an abbreviation, without necessarily indicating what the abbreviation may stand for. The <ex> and <expan> elements should be used where it is wished to signify that the content of the element is not present in the source but has been supplied by the transcriber, without necessarily indicating the abbreviation used in the original. The decision as to which course of action is appropriate may vary from abbreviation to abbreviation; there is no

requirement that the one system be used throughout a transcription, although doing so will generally simplify processing. The choice is likely to be a matter of editorial policy. If the highest priority is to transcribe the text literatim, while indicating the presence of abbreviations, the choice will be to use <abbr> or <am> throughout. If the highest priority is to present a reading transcription, while indicating that some letters or words are not actually present in the original, the choice will be to use <ex> or <expan> throughout.

Further information may be attached to instances of these elements by the <note> element, on which see section *3.8. Notes, Annotation, and Indexing*, and by use of the **resp** and **cert** attributes. In this instance from the English *Brut*, a note is attached to an editorial expansion of the tail on the final d of *good* to *goode*:

```
For alle the while that I had
good<ex xml:id="exp01">e</ex>
I was welbeloued
```

Source: [120]

Then the note:

```
<note target="#exp01">The stroke added to the final d could signify the
plural ending (-es, -is, -ys) but the singular <hi rend="it">good</hi> was used with the meaning
<q>property</q>,
<q>wealth</q>, at this time (v. examples quoted in OED, sb. Good,
C. 7, b, c, d and 8 spec.)</note>
```

The editor might declare a degree of certainty for this expansion, based on the OED examples, and state the responsibility for the expansion:

```
For alle the while that I had
good<ex resp="#mp" cert="high">e</ex> I was welbeloued
```

The value supplied for the **resp** attribute should point to the name of the editor responsible for this and possibly other interventions; an appropriate element therefore might be a <respStmt> element in the header like the following:

```
<respStmt xml:id="mp">
  <resp>Editorial emendations</resp>
  <name>Malcom Parkes</name>
</respStmt>
```

Observe that the **cert** and **resp** attributes are used with the <ex> element only to indicate confidence in the content of the element (i.e. the expansion), and responsibility for suggesting this expansion respectively.

The <choice> element may be used to indicate that the proposed expansion is one way of encoding what might equally well be represented as an abbreviation, represented by the hooked D, as follows:

```
For alle the while that I had
<choice>
  <sic>good<abbr>ꝺ</abbr>
  </sic>
  <expan resp="#mp" cert="high">good<ex>e</ex>
  </expan>
</choice>
I was welbeloued
```

If it is desired to express aspects of certainty and responsibility for some other aspect of the use of these elements, then the mechanisms discussed in chapter *21. Certainty and Responsibility* should be used. See also *11.4.2. Hand, Responsibility, and Certainty Attributes* for discussion of the issues of certainty and responsibility in the context of transcription.

If more than one expansion for the same abbreviation is to be recorded, multiple notes may be supplied. It may also be appropriate to use the markup for critical apparatus; an example is given in section *12.3. Using Apparatus Elements in Transcriptions*.

11.3.3 Correction and Conjecture

The <sic>, <corr>, and <choice> elements, defined in the core module should be used to indicate passages deemed in need of correction, or actually corrected, during the transcription of a source. For example, in the manuscript of William James's *A Pluralistic Universe*, edited by Fredson Bowers (Cambridge: Harvard University Press, 1977) a sentence first written

> One must have lived longer with this system, to appreciate its advantages.

has been modified by James to begin 'But One must ...', without the inital capital O having been reduced to lowercase. This non-standard orthography could be recorded thus:

```
But <sic>One</sic>
must have lived ...
```

or corrected:

```
But <corr>one</corr> must
have lived ...
```

or the two possibilities might be represented as a choice:

```
But
<choice>
  <sic>One</sic>
  <corr>one</corr>
</choice> must have lived
...
```

Similarly, in this example from Albertus Magnus, both a manuscript error *angues* and its correction *augens* are registered within a <choice> element:

```
Nos autem iam ostendimus quod nutrimentum
et <choice>
  <sic>angues</sic>
  <corr>augens</corr>
</choice>.
```

Source: [53]

Note that the <corr> element is used to provide a corrected form which is *not* present in the source; in the case of a correction made in the source itself, whether scribal, authorial, or by some other hand, the <add>, <del>, and <subst> elements described in *11.3.4. Additions and Deletions* should be used.

The <sic> element is used to mark passages considered by the transcriber to be erroneous; in such cases, the <corr> element indicates the transcriber's correction of them. Where the transcriber considers that one or more words have been erroneously omitted in the original source and corrects this omission, the <supplied> element discussed in *11.3.7. Text Omitted from or Supplied in the Transcription* should be used in preference to <corr>. Thus, in the following example, from George Moore's draft of additional materials for *Memoirs of My Dead Life*, the transcriber supplies the word *we* omitted by the author:

```
You see that I avoid the word create for we
create nothing <supplied>we</supplied> develope.
```

Source: [144]

As with <expan> and <abbr>, the choice as to whether to record simply that there is an apparent error, or simply that a correction has been applied, or to record both possible readings within a <choice> element is left to the encoder. The decision is likely to be a matter of editorial policy, which might be applied consistently throughout or decided case by case. If the highest priority is to present an uncorrected transcription while noting perceived errors in the original, the choice will typically be to use only <sic> throughout. If the highest priority is to present a reading transcription, while indicating that perceived errors in the original have been corrected, the choice will be to use only <corr> throughout.

Further information may be attached to instances of these elements by the <note> element and **resp** and **cert** attributes. Instances of these elements may also be classified according to any convenient typology using the **type** attribute.

For example, consider the following encoding of an emendation in the Hengwrt manuscript proposed by E. Talbot Donaldson:

```
Telle me also, to what conclusioun
Were membres maad, of generacioun
And of so parfit wis a
<choice  xml:id="corr117">
  <sic>wight</sic>
  <corr>wright</corr>
</choice>
ywroght?

<!-- ... -->
<note  target="#corr117">This emendation of the Hengwrt copy text,
based on a Latin source and on the reading of three late
and usually unauthoritative manuscripts, was proposed
by E. Talbot Donaldson in <bibl>
   <title>Speculum</title> 40 (1965)
    626–33.</bibl>
</note>
```

The <note> element discussed in *3.8. Notes, Annotation, and Indexing* may be used to give a more detailed discussion of the motivation for or scope of a correction. If linked by means of a pointer (as in this example) it may be located anywhere convenient within the transcription; typically all detailed notes will be collected together in a separate element in the <back>. Alternatively, the pointer may be omitted, and the <note> placed immediately adjacent to the element being annotated. The advantage of the former solution is that it permits the same annotation to refer to several corrections.

The attribute **cert** may be used to indicate the degree of confidence ascribed by the encoder to the proposed emendation on a broad scale: high, medium, or low. The attribute **resp** is used to indicate who is responsible for

the proposed emendation. Its value is a pointer, which will typically indicate a <respStmt> or <name> element in the header of the transcribed document, but can point anywhere, for example to some online authority file. Using these two attributes, the <corr> element presented above might usefully be enhanced as follows:

```
<!-- somewhere in the header ... --><name xml:id="ETD">E Talbot Donaldson</name>
<!-- ... -->
And of so parfit wis a
<choice>
  <sic>wight</sic>
  <corr resp="#ETD" cert="medium">wright</corr>
</choice>
ywroght?
```

As remarked above, where the same annotation applies to several corrections, this may be represented by supplying multiple pointers on the note. Consider for example such corrections as the following, in Dudo of S. Quentin. Parkes cites two cases in this manuscript of the same phenomenon:

```
quamuis <choice xml:id="sic-1">
  <sic>mens</sic>
  <corr>iners</corr>
</choice> que nutu dei
gesta sunt ... unde esset uiriliter
<choice xml:id="sic-2">
  <corr>uegetata</corr>
  <sic>negata</sic>
</choice>
```

Source: [64]

which may be described as follows:

```
<note target="#sic-1 #sic-2">Substitution of a more familiar word which resembles
graphically what the scribe should be copying but which does not make
sense in the context.</note>
```

The **target** attribute on the <note> element indicates the <choice> elements which exemplify this kind of scribal error. This necessitates the addition of an identifier to each <choice> element. However, if the number of corrections is large and the number of notes is small, it may well be both more practical and more appropriate to regard the collection of annotations as constituting a typology and then use the **type** attribute. Suppose that the note given above is one of half a dozen possible kinds of corrected phenomena identified in a given text; others might include, say, 'repetition of a word from the preceding line', etc. The **type** attribute on the <corr> element can be used to specify an arbitrary code for the particular kind of correction (or other editorial intervention) identified within it. This code can be chosen freely and is not treated as a pointer.

```
quamuis
<choice>
  <sic>mens</sic>
  <corr type="graphSubs">iners</corr>
</choice> que nutu dei
gesta sunt ... unde esset uiriliter
```

```
<choice>
  <corr  type="graphSubs">uegetata</corr>
  <sic>negata</sic>
</choice>
```

Note that this encoding might be extended to include a range of possible corrections:

```
quamuis
<choice>
  <sic>mens</sic>
  <corr  type="graphSubs">iners</corr>
  <corr  type="reversal">inres</corr>
</choice> que nutu dei
gesta sunt ...
```

In addition, the conscientious encoder will provide documentation explaining the circumstances in which particular codes are judged appropriate. A suitable location for this might be within the <correction> element of the <encodingDesc> of the header, which might include a <list> such as the following:

```
<correction>
  <p>The following codes are used to categorise corrections identified
     in this transcription:
  <list  type="gloss">
      <label>graphSubs</label>
      <item>Substitution of a more familiar word which resembles
            graphically what the scribe should be copying but which does not make
            sense in the context.</item>
<!-- ... -->
    </list>
  </p>
</correction>
```

A **subtype** attribute may be used in conjunction with the **type** for subclassification purposes: the above examples might thus be represented as **<choice type="substitution" subtype="graphicResemblence">** for example.

For a given project, it may well be desirable to limit the possible values for the **type** or **subtype** attributes automatically. This is easily done but requires customization of the TEI system using techniques described in *23.2. Personalization and Customization*, in particular *23.2.1.4. Modification of Attribute and Attribute Value Lists*, which should be consulted for further information on this topic.

When making a correction in a source which forms part of a textual tradition attested by many witnesses, a textual editor will sometimes use a reading from one witness to correct the reading of the source text. In the general case, such encoding is best achieved with the mechanisms provided by the module for textual criticism described in chapter *12. Critical Apparatus*. However, for simple cases, the **source** attribute of the <corr> attribute may suffice. In the passage from Chaucer's *Wife of Bath's Tale* mentioned above, Parkes proposes to emend the problematic word *wight* to *wyf* which is the reading found in the Cambridge manuscript Gg.1. 27. This may be simply represented as follows:

```
And of so parfit wis a
<choice>
```

```
  <sic>wight</sic>
  <corr  resp="#mp"  source="#Gg">wyf</corr>
</choice>
ywroght?
```

The value of the **source** attribute here is, like the value of the **resp** attribute, a pointer, in this case indicating the manuscript used as a witness. Elsewhere in the transcribed text, a list of witnesses used in this text will be given, one of which has an identifier `Gg`. Each witness will be represented either by a <witness> element (see *12.1. The Apparatus Entry, Readings, and Witnesses*) or more fully by a <msDesc> element (see *10. Manuscript Description*) :

```
<msDesc  xml:id="Gg">
  <msIdentifier>
    <settlement>Cambridge</settlement>
    <repository>University Library</repository>
    <idno>Gg.1. 27</idno>
  </msIdentifier>
<!-- further description of the manuscript here -->
</msDesc>
```

The <app> element described in chapter *12. Critical Apparatus* provides a more powerful way of representing all three possible readings in parallel:

```
And of so
parfit wis a
<app>
  <rdg  wit="#Hg">wight</rdg>
  <rdg  wit="#Ln #Ry2 #Ld">wright</rdg>
  <rdg  wit="#Gg">wyf</rdg>
</app>
```

This encoding simply records the three readings found in the various traditions, and gives (by means of the **wit** attribute) an indication of the witnesses supporting each. If the **resp** attribute were supplied on the <rdg> element, it would indicate the person responsible for asserting that the manuscript indicated has this reading, who is not necessarily the same as the person responsible for asserting that this reading should be used to correct the others. Editorial intervention elements such as <corr> can however be nested within a <rdg> to provide this additional information:

```
And of so
parfit wis a
<app>
  <rdg  wit="#Hg">wight</rdg>
  <rdg  wit="#Ln #Ry2 #Ld">
    <corr  resp="#ETD">wright</corr>
  </rdg>
  <rdg  wit="#Gg">
    <corr  resp="#mp">wyf</corr>
  </rdg>
</app>
```

This encoding asserts that the reading *wyf* found in Gg is regarded as a correction by Parkes.

Like the **resp** attribute, the **cert** attribute may be used with both <corr> and <rdg> elements. When used on the <rdg> element, these attributes indicate confidence in and responsibility for identifying the reading within the sources specified; when used on the <corr> element they indicate confidence in and responsibility for the use of the reading to correct the base text. If no other source is indicated (either by the **source** attribute, or by the **wit** attribute of a parent <rdg>), the reading supplied within a <corr> has been provided by the person indicated by the **resp** attribute.

If it is desired to express aspects of certainty and responsibility for some other aspect of the use of these elements, then the mechanisms discussed in chapter *21. Certainty and Responsibility* may be found useful. See also *11.4.2. Hand, Responsibility, and Certainty Attributes* for further discussion of the issues of certainty and responsibility in the context of transcription.

11.3.4 Additions and Deletions

Additions and deletions observed in a source text may be described using the following elements:

<add> (addition) contains letters, words, or phrases inserted in the text by an author, scribe, annotator, or corrector.

<addSpan/> (added span of text) marks the beginning of a longer sequence of text added by an author, scribe, annotator or corrector (see also «<add>»).

<del> (deletion) contains a letter, word, or passage deleted, marked as deleted, or otherwise indicated as superfluous or spurious in the copy text by an author, scribe, annotator, or corrector.

<delSpan/> (deleted span of text) marks the beginning of a longer sequence of text deleted, marked as deleted, or otherwise signaled as superfluous or spurious by an author, scribe, annotator, or corrector.

Of these, <add> and <del> are included in the core module, while <addSpan> and <delSpan> are available only when using the module defined in this chapter. These particular elements are members of the att.spanning class, from which they inherit the following attribute:

att.spanning provides attributes for elements which delimit a span of text by pointing mechanisms rather than by enclosing it.

@**spanTo** indicates the end of a span initiated by the element bearing this attribute.

Further characteristics of each addition and deletion, such as the hand used, its effect (complete or incomplete, for example), or its position in a sequence of such operations may conveniently be recorded as attributes of these elements, all of which are members of the att.transcriptional class:

att.transcriptional provides attributes specific to elements encoding authorial or scribal intervention in a text when transcribing manuscript or similar sources.

@**seq** (sequence) assigns a sequence number related to the order in which the encoded features carrying this attribute are believed to have occurred.

@**status** indicates the effect of the intervention, for example in the case of a deletion, strikeouts which include too much or too little text, or in the case of an addition, an insertion which duplicates some of the text already present.

@**hand** signifies the hand of the agent which made the intervention.

As described in section *3.4. Simple Editorial Changes*, the <add> element is used to record any manuscript addition observed in the text, whether it is considered to be authorial or scribal. In the autograph manuscript of Max Beerbohm's *The Golden Drugget*, the author's addition of *do ever* may be recorded as follows, with the **hand** attribute indicating that the addition was Beerbohm's by referencing a <handNote> element defined elsewhere in the document (see further *11.4.1. Document Hands*):

```
Some things are best at first
sight. Others – and here is one of them – <add  hand="#mb">do
ever</add> improve by recognition ....

<handNote  xml:id="mb">Max Beerbohm holograph</handNote>
```

Source: [12]

Similarly, when the <del> element is used to record manuscript deletions. In the autograph manuscript of D. H. Lawrence's *Eloi, Eloi, lama sabachthani* the author's deletion of *my* may be recorded as follows. In this case, the **hand** attribute indicating that the deletion was Lawrence's is complemented by a **rend** attribute indicating that the deletion was by strike-through:

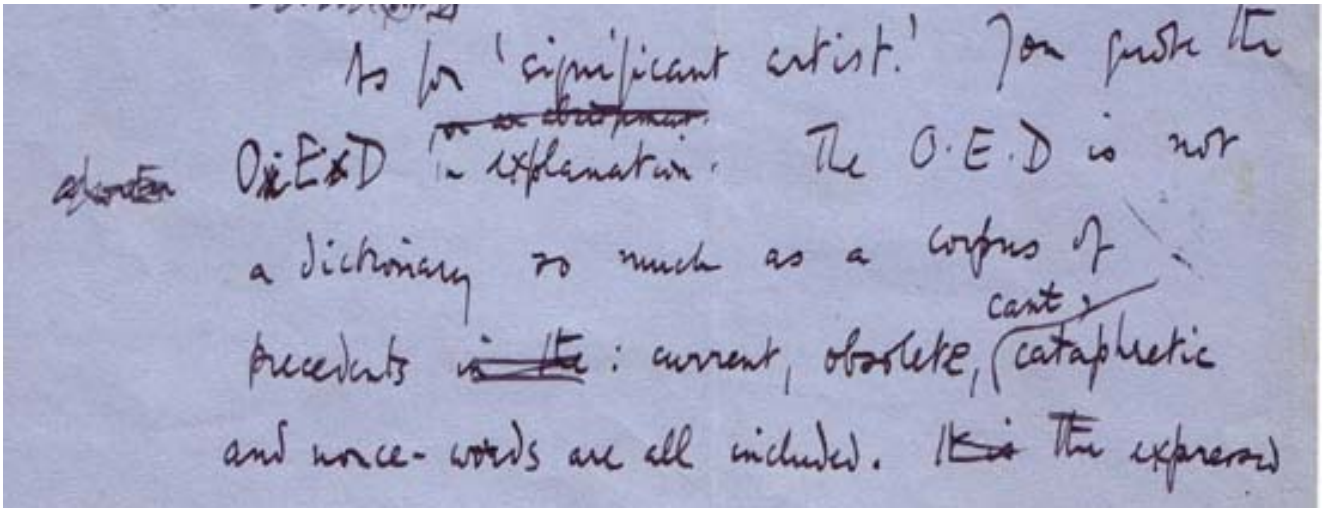
```
For I hate this <del  rend="strikethrough"  hand="#dhl">my</del> body, which is so dear to me
...

<handNote  xml:id="dhl">D H Lawrence holograph</handNote>
```

Source: [119]

If deletions are classified systematically, the **type** attribute may be useful to indicate the classification; when they are classified by the manner in which they were effected, or by their appearance, however, this will lead to a certain arbitrariness in deciding whether to use the **type** or the **rend** attribute to hold the information. In general, it is recommended that the **rend** attribute be used for description of the appearance or method of deletion, and that the **type** attribute be reserved for higher level or more abstract classifications.

The **place** attribute is also available to indicate the location of an addition. For example, consider the following passage from a draft letter by Robert Graves:



At the end of this extract, the writer inserts the word 'cant,' above the line, with a stroke to indicate insertion: this might be encoded as follows:

```
The O.E.D. is not a dictionary so much as a corpus of
precedents <del  hand="#RG">in the</del>: current,
obsolete, <add  hand="#RG"  place="supralinear">cant,</add>
cataphretic and nonce-words are all included.
```

Source: [90]

A little earlier in the same extract, Graves writes 'for an abridgement' above the line, and then deletes it. This may be encoded similarly:

```
As for 'significant artist.' You quote the O.E.D <add  hand="#RG"  place="supralinear">
  <del>for an abridgement</del>
</add>in
explanation...
```

Source: [90]

Similarly, in the margin, the word 'Norton' has been added and then deleted:

```
You quote the <add  hand="#RG"  place="left">
  <del>Norton</del>
</add>
O.E.D...
```

Source: [90]

The word 'O.E.D.' in this first sentence has also clearly been the result of some redrafting: it may be that Graves started to write 'Oxford', and then changed it; it may be that he inserted other punctuation marks between the letters before replacing them with the centre dots used elsewhere to represent this acronym. We do not deal with these possibilities here, and mention them only to indicate that any encoding of manuscript material of this complexity will need to make decisions about what is and is not worth mentioning.

An encoder may also wish to indicate that an addition replaces a specific deletion, that is to encode a substitution as a single intervention in the text. This may be achieved by grouping the addition and deletion together within a <subst> element. At the end of the passage illustrated above, Graves first writes 'It is the expressed...', then deletes 'It is', and substitutes an uppercase T at the start of 'the'.

```
...
are all included. <del  hand="#RG">It is</del>
<subst>
  <add>T</add>
  <del>t</del>
</subst>he expressed
```

Source: [90]

The use of this element and of the **seq** attribute to indicate the order in which interventions such as deletions are believed to have occurred are further discussed in section *11.3.5. Substitutions* below.

The <add> and <del> elements defined in the core module suffice only for the description of additions and deletions which fit within the structure of the text being transcribed, that is, which each deletion or addition is completely contained by the structural element (paragraph, line, division) within which it occurs. Where this is not the case, for example because an individual addition or deletion involves several distinct structural subdivisions, such as poems or prose items, or otherwise crosses a structural boundary in the text being encoded, special treatment is needed. The <addSpan> and <delSpan> elements are provided by this module for that purpose. (For a general discussion of the issue see further *20. Non-hierarchical Structures*).

In this example of the use of <addSpan>, the insertion by Helgi Ólafsson of a gathering containing four neo-Eddic poems into *Lbs 1562 4to* is recorded as follows.

A <handNote> element is first declared, within the header of the document, to associate the identifier **heol** with Helgi. Each of the added poems is encoded as a distinct element. In the body of the text,

an <addSpan> element is placed to mark the beginning of the span of added text, and an <anchor> is used to mark its end. The **hand** attribute on the <addSpan> element ascribes responsibility for the addition to the manuscript to Helgi, and the **spanTo** attribute points to the end of the added text:

```
<handNote xml:id="heol" scribe="HelgiÓlafsson"/>
<!-- ... -->
<body>
  <div>
<!-- text here -->
  </div>
  <addSpan n="added gathering" hand="#heol" spanTo="#p025"/>
  <div>
<!-- text of first added poem here -->
  </div>
  <div>
<!-- text of second added poem here -->
  </div>
  <div>
<!-- text of third added poem here -->
  </div>
  <div>
<!-- text of fourth added poem here -->
  </div>
  <anchor xml:id="p025"/>
  <div>
<!-- more text here -->
  </div>
</body>
```

Source: [66]

The <delSpan> element is used in the same way. An authorial manuscript will often contain several occasions where sequences of whole lines are marked for deletion, either by boxes or by being struck out. If the encoder is marking up individual verse lines with the <l> element, such deletions are problematic: deletion of two consecutive lines should be regarded as a single deletion, but the <del> element must be properly nested within a single <l> element. The <delSpan> element solves this problem:

```
<l>Flowed up the hill and down King William Street,</l>
<delSpan spanTo="#EPdelEnd" resp="#EP" rend="strikethrough"/>
<l>To where Saint Mary Woolnoth kept the time,</l>
<l>With a dead sound on the final stroke of nine.</l>
<anchor xml:id="EPdelEnd"/>
<l>There I saw one I knew, and stopped him, crying "Stetson!</l>...
```

It is also often the case that deletions and additions may themselves contain other deletions and additions. For example, in Thomas Moore's autograph of the second version of *Lalla Rookh* two lines are marked for omission by vertical strike-through. Within the first of the two lines, the word *upon* has also been struck out, and the word *over* has been added:

```
<l>
  <delSpan rend="verticalStrike" spanTo="#delend01"/>
Tis moonlight <del>upon</del>
```

```
  <add>over</add> Oman's sky
</l>
<l>Her isles of pearl look lovelily<anchor  xml:id="delend01"/>
</l>
```

Source: [145]

In this case the <anchor> and <delSpan> have been placed within the structural elements (the <l>s) rather than between, as in the previous example. This is to indicate that placement of these empty elements is arbitrary.

The text deleted must be at least partially legible, in order for the encoder to be able to transcribe it. If all of part of it is not legible, the <gap> element should be used to indicate where text has not not transcribed, because it could not be. The <unclear> element described in section *11.5.1. Damage, Illegibility, and Supplied Text* may be used to indicate areas of text which cannot be read with confidence. See further section *11.3.7. Text Omitted from or Supplied in the Transcription* and section *11.5.1. Damage, Illegibility, and Supplied Text*.

11.3.5 Substitutions

Substitution of one word or phrase for another is perhaps the most common of all phenomena requiring special treatment in transcription of primary textual sources. It may be simply one word overwriting another, or deletion of one word and its replacement by another written above it by the same hand at the one time; the deletion and replacement may be done by different hands at different times; there may be a long chain of substitutions on the one stretch of text, with uncertainty as to the order of substitution and as to which of many possible readings should be preferred.

As we have shown, the simplest method of recording a substitution is simply to record both the addition and the deletion. However, when the module defined by this chapter is in use, an additional element is available to indicate that the encoder believes the addition and the deletion to be part of the same intervention: a substitution.

<subst> (substitution) groups one or more deletions with one or more additions when the combination is to be regarded as a single intervention in the text.

Using this element, the example at the end of the last section might be encoded as follows:

```
<l>
  <delSpan  rend="verticalStrike"  spanTo="#delend02"/>
Tis moonlight <subst>
   <del>upon</del>
   <add>over</add>
  </subst> Oman's sky
</l>
<l>Her isles of pearl look lovelily<anchor  xml:id="delend02"/>
</l>
```

Since the purpose of this element is solely to group its child elements together, the order in which they are presented is not significant. By convention, however, deletion precedes addition. This may be overridden by means of the **seq** attribute, which is of particular usefulness when a sequence of deletions and additions occurs.

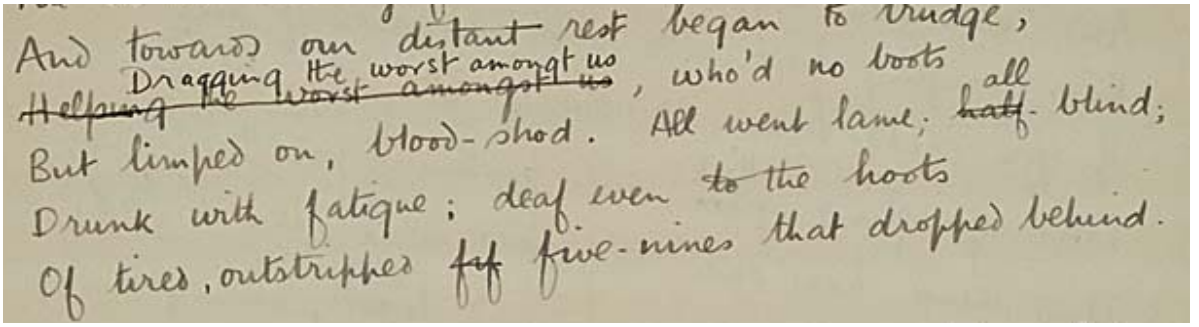
For example, returning to the example from William James, in a passage first written out by James as 'One must have lived longer with this system, to appreciate its advantages', the word *this* is first replaced by *such a* and *this* is then replaced by *a*.[4] This may be encoded as follows, representing the two changes as a sequence of additions and deletions:

[4] The manuscript contains several other substitutions, ignored here for the sake of clarity.

```
One must have lived longer
with <subst>
  <del  seq="1">this</del>
  <del  seq="2">
    <add  seq="1">such
        a</add>
  </del>
  <add  seq="2">a</add>
</subst> system, to appreciate its
advantages.
```

Note the nesting of an <add> element within a <del> to record text first added, then deleted in the source. The numbers assigned by the **seq** attribute may be used to identify the order in which the various additions and deletions are believed by the encoder to have been carried out, and thus provide a simple method of supporting the kind of 'genetic' textual criticism typified by (for example) Hans Walter Gabler's work on the reconstruction of the 'overlay' levels implicit in the manuscripts of James Joyce's *Ulysses*.

As a more complex example, consider the following passage in one of the manuscripts of Wilfred Owen's *Dulce et decorum est*:



This passage might be encoded as follows:

```
<l>And towards our distant rest began to trudge,</l>
<l>
  <subst>
    <del>Helping the worst amongst us</del>
    <add>Dragging the
        worst amongt us</add>
  </subst>, who'd no boots
</l>
<l>But limped on, blood-shod.  All went lame;
<subst>
    <del  status="shortEnd">half-</del>
    <add>all</add>
  </subst> blind;</l>
<l>Drunk with fatigue ; deaf even to the hoots</l>
<l>Of tired, outstripped <del>fif</del> five-nines that dropped behind.</l>
```

Source: [151]

In this representation,

- the false start *fif* in the last line is simply marked as a deletion;
- the other two authorial corrections are marked as substitutions, each combining a deletion and an addition.

- the authorial slip (*amongt* for *amongst*) is retained without comment.

The <app> element presented in chapter *12. Critical Apparatus* provides similar facilities, by treating each state of the text as a distinct reading. The <rdg> element has a **varSeq** attribute which may be used in the same way as the **seq** attribute to indicate the preferred sequence. The James example above might thus be represented as follows:

```
One must have lived longer with
<app>
  <rdg  varSeq="1">
    <del>this</del>
  </rdg>
  <rdg  varSeq="2">
    <del>
      <add>such a</add>
    </del>
  </rdg>
  <rdg  varSeq="3">
    <add>a</add>
  </rdg>
</app>
system, to appreciate its advantages.
```

11.3.6 Cancellation of Deletions and Other Markings

An author or scribe may mark a word or phrase in some way, and then on reflection decide to cancel the marking. For example, text may be marked for deletion and the deletion then cancelled, thus restoring the deleted text. Such cancellation may be indicated by the <restore> element:

<restore> indicates restoration of text to an earlier state by cancellation of an editorial or authorial marking or instruction.

This element bears the same attributes as the other transcriptional elements. These may be used to supply further information such as the hand in which the restoration is carried out, the type of restoration, and the person responsible for identifying the restoration as such, in the same way as elsewhere.

Presume that Lawrence decided to restore *my* to the phrase of *Eloi, Eloi, lama sabachthani* first written 'For I hate this my body', with the *my* first deleted then restored by writing 'stet' in the margin. This may be encoded:

```
For I hate this
<restore  hand="#dhl"  type="marginalStetNote">
  <del>my</del>
</restore>
body
```

Another feature commonly encountered in manuscripts is the use of circles, lines, or arrows to indicate transposition of material from one point in the text to another. No specific markup for this phenomenon is proposed at this time. Such cases are most simply encoded as additions at the point of insertion and deletions at the point of encirclement or other marking.

11.3.7 Text Omitted from or Supplied in the Transcription

Where text is not transcribed, whether because of damage to the original, or because it is illegible, or for some other reason such as editorial policy, the <gap> core element should be used to register the omission; where text not present in the source is supplied (whether conjecturally or from other witnesses) to fill an apparent gap in the text, it should be marked using the <supplied> element provided by the module defined in this chapter.

<gap> indicates a point where material has been omitted in a transcription, whether for editorial reasons described in the TEI header, as part of sampling practice, or because the material is illegible, invisible, or inaudible.

@**reason** gives the reason for omission. Sample values include **sampling**, **illegible**, **inaudible**, **irrelevant**, **cancelled**, **illegible**.

@**hand** in the case of text omitted from the transcription because of deliberate deletion by an identifiable hand, signifies the hand which made the deletion.

@**agent** In the case of text omitted because of damage, categorizes the cause of the damage, if it can be identified.

<supplied> signifies text supplied by the transcriber or editor for any reason, typically because the original cannot be read because of physical damage or loss to the original.

@**reason** indicates why the text has had to be supplied.

By its nature, the <gap> element has no content. It marks a point in the text where nothing at all can be read, whether because of authorial or scribal erasure, physical damage, or any other form of illegibility. Its attributes allow the encoder to specify the amount of text which is illegible in this way at this point, using any convenient units, where this can be determined. For example, in the Beerbohm manuscript of *The Golden Drugget* cited above, the author has erased a passage amounting about 10 cm in length by inking over it completely:

```
Others <gap
   reason="cancelled"
   hand="#mb"
   extent="10"
   unit="cm"/>—and
here is one of them...
```

Source: [12]

In an autograph letter of Sydney Smith now in the Pierpont Morgan library three words in the signature are quite illegible:

```
I am dr Sr yr <gap  reason="illegible"  extent="3"  unit="word"/>Sydney Smith
```

Source: [183]

The degree of precision attempted when measuring the size of a gap will vary with the purpose of the encoding and the nature of the material: no particular recommendation is made here.

As noted above, the <gap> element should only be used where text has not been transcribed; if partially legible text has been transcribed, one of the elements <damage> and <unclear> should be used instead. These elements are described in section *11.5.1. Damage, Illegibility, and Supplied Text*.

If the source text is completely illegible or missing, an encoder may sometimes wish to supply new (conjectural) material to replace it. This conjectural reading is analogous to a correction in that it contains text provided by the encoder and not attested in the source. This is not however a correction, since no error is necessarily present in the original; for that reason a different element <supplied> should be used. If another (imaginary) copy of the letter above preserved the signature as reading 'I am dear Sir your very humble Servt Sydney Smith', the text illegible in the autograph might be supplied in the transcription:

```
I
am dr Sr yr <supplied  reason="illegible"  resp="#msm"  source="#AmCo">very humble Servt</supplied> Sydney Smith
```

Here the **source** and **resp** attributes are used, as elsewhere, to indicate respectively the sigil of a manuscript from which the supplied reading has been taken, and the identifier of the person responsible for deciding to supply the text. If the **source** attribute is not supplied, the implication is that the encoder (or whoever is indicated by the value of the **resp** attribute) has supplied the missing reading. Both <gap> and <supplied> may be used in combination with <unclear>, <damage>, and other elements; for discussion, see section *11.5.2. Use of the <gap>, <del>, <damage>, <unclear>, and <supplied> Elements in Combination*.

11.4 Hands and Responsibility

This section discusses in more detail the representation of aspects of responsibility perceived or to be recorded for the writing of a primary source. These include points at which one scribe takes over from another, or at which ink, pen, or other characteristics of the writing change. A discussion of the usage of the **hand**, **resp**, and **cert** attributes is also included.

11.4.1 Document Hands

For many text-critical purposes it is important to signal the person responsible (the hand) for the writing of a whole document, a stretch of text within a document, or a particular feature within the document. A hand, as the name suggests, need not necessarily be identified with a particular known (or unknown) scribe or author; it may simply indicate a particular combination of writing features recognized within one or more documents. The examples given above of the use of the **hand** attribute with coding of additions and deletions illustrate this.

The <handNote> element is used to provide information about each hand distinguished within the encoded document.

<handNote> (note on hand) describes a particular style or hand distinguished within a manuscript.

A <handNote> element, with an identifier given by its **xml:id** attribute, may appear in either of two places in the TEI Header, depending on which modules are included in a schema. When the transcr module defined by the present chapter is used, the element <handNotes> is available, within the <profileDesc> element of the Header, to hold one or more <handNote> elements. When the msdescription module defined in chapter *10. Manuscript Description* is included, the <handDesc> element described in *10.7.2. Writing, Decoration, and Other Notations* also becomes available as part of a structured manuscript description. The encoder may choose to place <handNote> elements identifying individual hands in either location without affecting their accessibility since the element is always addressed by means of its **xml:id** attribute. The <handDesc> element may be more appropriate when a full cataloguing of each manuscript is required; the <handNotes> element if only a brief characterization of each hand is needed. It is also possible to use the two elements together if, for example, the <handDesc> element contains a single summary describing all the hands discursively, while the <handNotes> element gives specific details of each. The choice will depend on individual encoders' priorities.

As shown above, the **hand** attribute is available on several elements to indicate the hand in which the content of the element (usually a deletion or addition) is carried out. The <handShift> element may also be used within the body of a transcription to indicate where a change of hand is detected for whatever reason.

<handShift/> marks the beginning of a sequence of text written in a new hand, or the beginning of a scribal stint.

Both <handShift> and <handNote> are members of the att.handFeatures class, and thus share the following attributes:

att.handFeatures provides attributes describing aspects of the hand in which a manuscript is written.

- **@scribe** gives a standard name or other identifier for the scribe believed to be responsible for this hand.
- **@script** characterizes the particular script or writing style used by this hand, for example *secretary*, *copperplate*, *Chancery*, *Italian*, etc.

@**medium** describes the tint or type of ink, e.g. *brown*, or other writing medium, e.g. *pencil*

@**scope** specifies how widely this hand is used in the manuscript.

A single hand may employ different writing styles and inks within a document, or may change character. For example, the writing style might shift from 'anglicana' to 'secretary', or the ink from blue to brown, or the character of the hand may change. Simple changes of this kind may be indicated by assigning a new value to the appropriate attribute within the <handShift> element. It is for the encoder to decide whether a change in these properties of the writing style is so marked as to require treatment as a distinct hand.

Where such a change is to be identified, the **new** attribute is used to indicate the hand applicable to the material following the <handShift>. This will ordinarily, but not necessarily, be the order in which the material was originally written.

As might be expected, one hand may employ different renditions within the one writing style, for example medieval scribes often indicate a structural division by emboldening all the words within a line. These should be indicated by use of the **rend** attribute on an element, in the same manner as underlining, emboldening, font shifts, etc. are represented in transcription of a printed text, rather than by introducing a new <handShift> element.

In the following example there is a change of ink within the one hand. This is simply indicated by a new value for the **medium** attribute on the <handShift> element:

```
<l>When wolde the cat dwelle in his ynne</l>
<handShift medium="greenish-ink"/>
<l>And if the cattes skynne be slyk <handShift medium="black-ink"/> and gaye</l>
```

Source: [34]

In the following example, the encoder has identified two distinct hands within the document and given them identifiers **h1** and **h2**, by means of the following declarations included in the document's TEI Header:

```
<handNotes>
  <handNote xml:id="h1" script="copperplate" medium="brown-ink">Carefully written with regular descen-
ders</handNote>
  <handNote xml:id="h2" script="print" medium="pencil">Unschooled scrawl</handNote>
</handNotes>
```

Then the change of hand is indicated in the text:

```
<handShift new="#h1" resp="#das"/>... and that good Order Decency and regular worship
may be once more introduced and Established in this
Parish according to the Rules and Ceremonies of the
Church of England and as under a good Consciencious
and sober Curate there would and ought to be
<handShift new="#h2" resp="#das"/>
and for that purpose the parishioners pray
```

Source: [52]

11.4.2 Hand, Responsibility, and Certainty Attributes

The **hand** and **resp** attributes have similar, but not identical, meanings. Observe their distinctive uses in the following encoding of the William James passage mentioned above in section *11.3.3. Correction and Conjecture*. In this example, the *But* inserted by James is tagged as an <add>, and the consequent editorial correction of *One* to *one* treated separately:

```
<add place="supralinear" resp="#FB" hand="#WJ">But</add>
<choice>
  <sic>One</sic>
  <corr resp="#FB">one</corr>
</choice> must have
lived ...

<!-- elsewhere -->
<respStmt xml:id="FB">
  <resp>editorial changes</resp>
  <name>Fredson Bowers</name>
</respStmt>
<respStmt xml:id="WJ">
  <resp>authorial changes</resp>
  <name>William James</name>
</respStmt>
```

As in this example, **hand** should be reserved for indicating the hand of any form of marking—here, addition but also deletion, correction, annotation, underlining, etc.—within the primary text being transcribed. The scribal or authorial responsibility for this marking may be inferred from the value of the **hand** attribute. The value of the **hand** attribute should be one of the hand identifiers declared in the document header (see section *11.4.1. Document Hands*).

The **resp** attribute, by contrast, indicate the person responsible for deciding to apply the element carrying it to this part of the text, and hence has a slightly different interpretation. In the case of the <add> element, for example, the **resp** attribute will indicate the responsibility for identifying that the addition is indeed an addition, and also (if the **hand** attribute is supplied) to which hand it should be attributed. In this case, Bowers is credited with identifying the hand as that of William James. In the case of the <corr> element, the **resp** attribute indicates who is responsible for supplying the intellectual content of the correction reported in the transcription: here, Bowers' correction of 'One' to 'one'. In the case of a deletion, the **resp** attribute will similarly indicate who bears responsibility for identifying or categorising the deletion itself, while other attributes (**hand** most obviously) attribute responsibility for the deletion itself.

As these examples show, the field of application of the **resp** attributes varies from element to element. In some cases, it applies to the content of the element (<corr>, <ex>, and <supplied>); in others it applies to the value of a particular attribute (<sic>, <abbr>, <del>, etc.). In all cases where both the **resp** and **cert** attributes are defined for a particular element, the two attributes refer to the same aspect of the markup. The one indicates who is intellectually responsible for some item of information, the other indicates the degree of confidence in the information. Thus, for a correction, the **resp** attribute signifies the person responsible for supplying the correction, while the **cert** attribute signifies the degree of editorial confidence felt in that correction. For the expansion of an abbreviation, the **resp** attribute signifies the person responsible for supplying the expansion and the **cert** attribute signifies the degree of editorial confidence felt in the expansion.

This close definition of the use of the **resp** and **cert** attributes with each element is intended to provide for the most frequent circumstances in which encoders might wish to make unambiguous statements regarding the responsibility for and certainty of aspects of their encoding. The **resp** and **cert** attributes, as so defined, give a convenient mechanism for this. However, there will be cases where it is desired to state responsibility for and certainty concerning other aspects of the encoding. For example, one may wish in the case of an apparent addition to state the responsibility for the use of the <add> element, rather than the responsibility for identifying the hand of the addition. It may also be that one editor may make an electronic transcription of another editor's printed transcription of a manuscript text — here, one will wish to assign layers of responsibility, so as to allow the reader to determine exactly what in the final transcription was the responsibility of each editor. In these complex cases of divided editorial responsibility for and certainty concerning the content, attributes, and

application of a particular element, the more general mechanisms for representing certainty and responsibility described in chapter *21. Certainty and Responsibility* should be used.

It should be noted that the certainty and responsibility mechanisms described in chapter *21. Certainty and Responsibility* replicate all the functions of the **resp** and **cert** attributes on particular elements. For example, the encoding of Donaldson's conjectured emendation of *wight* to *wright* in line 117 of Chaucer's *Wife of Bath's Prologue* (see *11.3.3. Correction and Conjecture*) may be encoded as follows using the **resp** and **cert** attributes on the <corr> element:

```
<choice>
  <sic>wight</sic>
  <corr  resp="#ETD"  cert="medium">wright</corr>
</choice>
```

Exactly the same information could be conveyed using the certainty and responsibility mechanisms, as follows:

```
<choice>
  <corr  xml:id="c117">wright</corr>
  <sic>wight</sic>
</choice>
<certainty  target="#c117"  locus="transcribedContent"  degree="0.7"/>
<respons  target="#c117"  locus="transcribedContent"  resp="#ETD"/>
```

The choice of which mechanism to use is left to the encoder. In transcriptions where only such statements of responsibility and certainty are made as can be accommodated within the **resp** and **cert** attributes of particular elements, it will be economical to use the **resp** and **cert** attributes of those elements. Where many statements of responsibility and certainty are made which cannot be so accommodated, it may be economical to use the <respons> and <certainty> elements throughout.

The above discussion supposes that in each case an encoder is able to specify exactly what it is that one wishes to state responsibility for and certainty about. Situations may arise when an encoder wishes to make a statement concerning certainty or responsibility but is unable or unwilling to specify so precisely the domain of the certainty or responsibility. In these cases, the <note> element may be used with the **type** attribute set to 'cert' or 'resp' and the content of the note giving a prose description of the state of affairs.

11.5 Damage and Conjecture

The carrier medium of a primary source may often sustain physical damage which makes parts of it hard or impossible to read. In this section we discuss elements which may be used to represent such situations and give recommendations about how these should be used in conjunction with the other related elements introduced previously in this chapter.

11.5.1 Damage, Illegibility, and Supplied Text

The <gap> and <supplied> elements described above (section *11.3.7. Text Omitted from or Supplied in the Transcription*) should be used with appropriate attributes where the degree of damage or illegibility in a text is such that nothing can be read and the text must be either omitted or supplied conjecturally or from one or more other sources. In many cases, however, despite damage or illegibility, the text may yet be read with reasonable confidence. In these cases, the following elements should be used:

<damage> contains an area of damage to the text witness.

<damageSpan/> (damaged span of text) marks the beginning of a longer sequence of text which is damaged in some way but still legible.

As members of the class att.damaged, these elements bear the following attributes

att.damaged provides attributes describing the nature of any physical damage affecting a reading.

- @**extent** indicates approximately how much text is in the damaged area, in letters, minims, inches, or any appropriate unit, where this cannot be deduced from the contents of the tag.
- @**hand** In the case of damage (deliberate defacement, inking out, etc.) assignable to a distinct hand, signifies the hand responsible for the damage.
- @**agent** categorizes the cause of the damage, if it can be identified.
- @**degree** Signifies the degree of damage according to a convenient scale. The <damage> tag with the **degree** attribute should only be used where the text may be read with some confidence; text supplied from other sources should be tagged as <supplied>.
- @**group** assigns an arbitrary number to each stretch of damage regarded as forming part of the same physical phenomenon.

As a member of the att.spanning class, <damageSpan> inherits the following additional attribute:

att.spanning provides attributes for elements which delimit a span of text by pointing mechanisms rather than by enclosing it.

- @**spanTo** indicates the end of a span initiated by the element bearing this attribute.

The following examples all refer to the recto of folio 5 of the unique manuscript of the Elder Edda. Here, the manuscript of *Vóluspá* has been damaged through irregular rubbing so that letters in various places are obscured and in some cases cannot be read at all.

In the first line of this leaf, the transcriber may believe that the last three letters of *daga* can be read clearly despite the damage:

```
um aldr
d<damage>aga</damage> yndisniota
```

Source: [203]

If, as is often the case, the damage crosses structural divisions, so that the <damage> element cannot be nested properly within the containing elements, the <damageSpan> element may be used, in the same way as the <delSpan> and <addSpan> elements discussed in section *11.3.4. Additions and Deletions*.

```
<p>
<!-- ... -->
  <pb  n="5r"/>
  <damageSpan  agent="rubbing"  extent="whole leaf"  spanTo="#damageEnd"/>
</p>
<p> .... </p>
<p> ....
<pb  n="5v"  xml:id="damageEnd"/>
</p>
```

Source: [203]

Note that in this example the **spanTo** element points to the next <pb> element rather than to an inserted <anchor> element, since the whole of the leaf (the text between the two <pb> elements has sustained damage. For other techniques of handling non-nesting information, see chapter *20. Non-hierarchical Structures*.

If, as is also likely, the damage affects several disjoint parts of the text, each such part must be marked with a separate <damage> or <damageSpan> element. To indicate that each of these is to be regarded as forming part

of the same damaged area, the **group** attribute may be used as in the following example. In this (imaginary) text of Fitzgerald's translation from Omar Khayam, water damage has affected an area covering parts of several lines

```
<l>The Moving Finger wri<damage agent="water" group="1">es; and</damage> having writ,</l>
<l>Moves <damage agent="water" group="1">on: nor all your</damage> Piety nor Wit</l>
<l>
  <damageSpan agent="water" group="1" spanTo="#washOut"/>Shall lure it back to cancel half a Line,
</l>
<l>Nor all your Tears wash <anchor xml:id="washOut"/> out a Word of it</l>
```

A more general solution to this problem is provided by the <join> element discussed in *16.7. Aggregation* which may be used to link together arbitrary elements of any kind in the transcription. Where, as here, several phenomena of illegibility and conjecture all result from the one cause, an area of damage to the text — rubbing at various points — which is not continuous in the text, affecting it at irregular points, the <join> element may be used to indicate which tagged features are part of the same physical phenomenon.

If the damage has been so severe as to render parts of the text only imperfectly legible, the <unclear> element should be used to mark the fact. Returning to the Eddic example above, an encoder less confident in the *daga* reading, may indicate this as follows:

```
um aldr d<unclear reason="damage">aga</unclear> yndisniota
```

If it is desired to supply more information about the kind of damage, it is also possible to nest an <unclear> element within the <damage> element:

```
um aldr d<damage agent="rubbing">
  <unclear>aga</unclear>
</damage> yndisniota
```

Alternatively, the transcriber may not feel able to read the last three letters of *daga* but may wish to supply them by conjecture. Note the use of the **resp** attribute to assign the conjecture to Finnur Jónsson:

```
um aldr d<supplied reason="rubbing" resp="#FJ">aga</supplied> yndisniota
```

Source: [203]

The <supplied> element may if desired be enclosed within a <damage> element:

```
um aldr d<damage agent="rubbing">
  <supplied source="#FJ">aga</supplied>
</damage> yndisniota
```

Source: [203]

Contrast the use of <gap> in the next line, where the transcriber believes that four letters cannot be read at all because of the damage:

```
þar komr inn dimmi dreki fliugandi naþr frann
neþan <gap
    reason="illegible"
    agent="rubbing"
    extent="4"
    unit="letters"/>
```

Source: [203]

As with <supplied>, this <gap> might be enclosed by a <damage> element.

The above examples record imperfect legibility due to damage. When imperfect legibility is due to some other reason (typically because the handwriting is ill-formed), the <unclear> element should be used without any enclosing <damage> element. In Robert Southey's autograph of *The Life of Cowper* the final six letters of *attention* are difficult to read because of the haste of the writing, though reasonably certain from the context.

```
and from time to time invited in like manner
his att<unclear>ention</unclear>
```

Source: [184]

The **cert** attribute on the <unclear> element may be used to indicate the level of editorial confidence in the reading contained within it.

11.5.2 Use of the <gap>, <del>, <damage>, <unclear>, and <supplied> Elements in Combination

The <gap>, <damage>, <unclear>, <supplied>, and <del> elements may be closely allied in their use. For example, an area of damage in a primary source might be encoded with any one of the first four of these elements, depending on how far the damage has affected the readability of the text. Further, certain of the elements may nest within one another. The examples given in the last sections illustrate something of how these elements are to be distinguished in use. This may be formulated as follows:

- where the text has been rendered completely illegible by deletion or damage and no text is supplied by the editor in place of what is lost: place an empty <gap> element at the point of deletion or damage. Use the **reason** attribute to state the cause (damage, deletion, etc.) of the loss of text.
- where the text has been rendered completely illegible by deletion or damage and text is supplied by the editor in place of what is lost: surround the text supplied at the point of deletion or damage with the <supplied> element. Use the **reason** attribute to state the cause (damage, deletion, etc.) of the loss of text leading to the need to supply the text.
- where the text has been rendered partly illegible by deletion or damage so that the text can be read but without perfect confidence: transcribe the text and surround it with the <unclear> element. Use the **reason** attribute to state the cause (damage, deletion, etc.) of the uncertainty in transcription and the **cert** attribute to indicate the confidence in the transcription.
- where there is deletion or damage but at least some of the text can be read with perfect confidence: transcribe the text and surround it with the <del> element (for deletion) or the <damage> element (for damage). Use appropriate attribute values to indicate the cause and type of deletion or damage. Observe that the **degree** attribute on the <damage> element permits the encoding to show that a letter, word, or phrase is not perfectly preserved, though it may be read with confidence.
- where there is an area of deletion or damage and parts of the text within that area can be read with perfect confidence, other parts with less confidence, other parts not at all: in transcription, surround the whole area

with the <del> element (for deletion; or the <delSpan> element where it crosses a structural boundary); or the <damage> element (for damage). Text within the damaged area which can be read with perfect confidence needs no further tagging. Text within the damaged area which cannot be read with perfect confidence may be surrounded with the <unclear> element. Places within the damaged area where the text has been rendered completely illegible and no text is supplied by the editor may be marked with the <gap> element. For each element, one may use appropriate attribute values to indicate the cause and type of deletion or damage and the certainty of the reading.

The rules for combinations of the <add> and <del> elements, and for the interpretation of such combinations, are similar:

- if one <add> element (with identifier ADD1) contains another (with identifier ADD2), then the addition ADD1 was first made to the text, and later a second addition (ADD2) was made within that added text:

```
This is the text
<add  xml:id="ADD1">with some added
<add  xml:id="ADD2">(interlinear!)</add>
material</add>
as written.
```

- if one <del> element contains another, and the **seq** attribute does not indicate otherwise, it should be assumed that the inner deletion was made before the enclosing one. In the following example, the word *redundant* was deleted before a second second deletion removed the entire passage:

```
<del>This sentence contains
some <del>redundant</del> unnecessary
verbiage.</del>
```

- if a <del> element contains an <add> element, the normal interpretation will be that an addition was made within a passage which was later deleted in its entirety:

```
<del>This sentence was deleted
<add>originally</add> from the text.</del>
```

- if an <add> element contains a <del> element, the normal interpretation will be that a deletion was made from a passage which had earlier been added:

```
<add>This sentence was added
<del>eventually</del> to the text.</add>
```

11.6 Aspects of Layout

Finally in this chapter we present elements which may be used to capture aspects of the layout of material on a page where this is considered important. Methods for recording page breaks, column breaks, and line breaks in the source are described in section *3.10. Reference Systems*.

11.6.1 Space

The author or scribe may have left space for a word, or for an initial capital, and for some reason the word or capital was never supplied and the space left empty. The presence of significant space in the text being transcribed may be indicated by the <space> element.

<space/> indicates the location of a significant space in the copy text.

@**resp** (responsible party) indicates the individual responsible for identifying and measuring the space.

Note that this element should not be used to mark normal inter-word space or the like.

In line 694 of Chaucer's *Wife of Bath's Prologue* in the Holkham manuscript the scribe has left a space for a word where other manuscripts read *preestes*:

```
By god if wommen had writen storyes
As <space  quantity="7"  unit="char"/> han within her oratoryes
```

The <supplied> element discussed in the previous section may be used to supply the text presumed missing:

```
By god if wommen had writen storyes
As <supplied  reason="space"  resp="#ES"  source="#Hg">preestes</supplied>
han within her oratoryes
```

Here, the fact of the space within the manuscript is indicated by the value of the **reason** attribute. The source of the supplied text is shown by the value of the **source** attribute as the Hengwrt manuscript; the transcriber responsible for supplying the text is ES.

11.6.2 Lines

The most common form of marking of text in manuscripts is by lines written under, beside, or through the text. The lines themselves may be of various types: they may be solid, dashed or dotted, doubled or tripled, wavy or straight, or a combination of these and other renderings. The line may be used for emphasis, or to mark a foreign or technical term, or to signal a quotation or a title, etc.: the elements <emph>, <foreign>, <term>, <mentioned>, <title> may be used for these. Frequently, a scholar may judge that a line is used to delete text: the <del> element is available to indicate this. In all these cases, the **rend** attribute may be used on these or other elements to indicate that the text is marked by a line and the style of the line. Thus, Lawrence's deletion by strike-through of *my* in the autograph of *Eloi, Eloi, lama sabachthani* is noted:

```
For I hate this
<del  rend="strikethrough"  hand="#dhl">my</del> body,
which is so dear to me
```

There will be instances, however, where a scholar wishes only to register the occurrence of lines in the text, without making any judgement as to what the lines signify. In these the <hi> element may be used, with the **rend** attribute to mark the style of line. In the manuscript of a letter by Robert Browning to George Moulton-Barrett the underlining of the phrase *had obtained all the letters to Mr Boyd* may be marked-up as follows:

```
I have once – by declaring I would prosecute
by law – hindered a man's proceedings who
<hi  rend="underline">had obtained all the letters
to Mr Boyd</hi>
```

Source: [21]

The above examples presume the common case where a single word or phrase is marked by a line, with no doubt as to where the marking begins or ends and with no overlapping of the area of text with other marked areas of text. Where there is doubt, the <certainty> element may be used to record the doubt. In the Browning example cited above the underlining actually begins half-way under *who*, and this uncertainty could be remarked as follows:

```
I have once – by declaring I would prosecute
by law – hindered a man's proceedings who
<hi xml:id="cstart1" rend="underline">had obtained all
the letters to Mr Boyd</hi>
<!-- ... -->
<certainty target="#cstart1" locus="startLoc" degree="0.70">
  <desc>may begin with previous word</desc>
</certainty>
```

Where the area of text marked overlaps other areas of text, for example crossing a structural division, one of the spanning mechanisms mentioned above must be used; for example where the line is thought to mark a deletion, the <delSpan> element may be used. Where it is desired simply to record the marking of a span of text in circumstances where it is not possible to surround the text with a <hi> element, the element may be used with the **rend** or **type** attribute indicating the style of line-marking.

More work needs to be done on clarifying the treatment of other textual features marked by lines which might so overlap or nest. For example, in many Middle English manuscripts (e.g. the Jesus and Digby verse collections), marginal sidebars may indicate metrical structure: couplets may be linked in pairs, with the pairs themselves linked into stanzas. Or, marginal sidebars may indicate emphasis, or may point out a region of text on which there is some annotation: in many manuscripts of Chaucer's *Wife of Bath's Prologue* lines 655–8 are marked with nesting parentheses against which the scribe has written *nota*.

At the lowest level, all such features could be captured by use of the <note> element, containing a prose description of the manuscript at this point, enhanced by a link to a visual representation (or facsimile) of the feature in question. It is not yet clear how best to mark up such phenomena so as to obtain more usefully structured encodings. For example, in the Chaucer example just cited, one may wish to record that the *nota* is written in the Hengwrt manuscript in the right margin against a single large left parenthesis bracketing the four lines, with two right parentheses in the right margin bracketing two overlapping pairs of lines: the first and third, the second and fourth. The <note> element allows us to record that the scribe wrote *nota*, but is not well-adapted to show that the *nota* points both at all four lines and at two pairs of lines within the four lines.

11.7 Headers, Footers, and Similar Matter

As a rule, matter associated with the page break (signature, catchword, page number) should be drawn into the <pb> element as attributes: see section *3.10. Reference Systems*. In text-critical situations where these elements need tagging in their own right (for instance, when the catch-word presents a variant reading, or spacing in the header or footer is significant for compositor identification), the element <fw> may be used:

<fw> (forme work) contains a running head (e.g. a header, footer), catchword, or similar material appearing on the current page.

The name *fw* is short for 'forme work'. It may be used to encode any of the unchanging portions of a page forme, such as:

- running heads (whether repeated or changing on every page, or alternating pages)
- running footers
- page numbers
- catch-words
- other material repeated from page to page, which falls outside the stream of the text

It should not be used for marginal glosses, annotations, or textual variants, which should be tagged using <gloss>, <note>, or the text-critical tags described in chapter *12. Critical Apparatus*, respectively.

For example:

```
<fw type="head" place="top-centre">Poëms.</fw>
<fw type="pageNum" place="top-right">29</fw>
<fw type="sig" place="bot-centre">E3</fw>
<fw type="catch" place="bot-right">TEMPLE</fw>
```

11.8 Other Primary Source Features not Covered in these Guidelines

We repeat the advice given at the beginning of this chapter, that these recommendations are not intended to meet every transcriptional circumstance ever likely to be faced by any scholar. They are intended rather as a base to enable encoding of the most common phenomena found in the course of scholarly transcription of primary source materials. These guidelines particularly do not address the encoding of physical description of textual witnesses: the materials of the carrier, the medium of the inscribing implement, the organisation of the carrier materials themselves (as quiring, collation, etc.), authorial instructions or scribal markup, etc., except insofaras these are involved in the broader question of manuscript description, as addressed by the msdescription module described in chapter *10. Manuscript Description*.

11.9 Module for Transcription of Primary Sources

The module described in this chapter makes available the following components:

Module transcr: Transcription of primary sources

- Elements defined: addSpan am damage damageSpan delSpan ex facsimile fw handNotes handShift restore space subst supplied surface zone
- Classes defined: att.coordinated att.global.facs

The selection and combination of modules to form a TEI schema is described in *1.2. Defining a TEI Schema*.

Chapter 12

Critical Apparatus

Scholarly editions of texts, especially texts of great antiquity or importance, often record some or all of the known variations among different witnesses to the text. Witnesses to a text may include authorial or other manuscripts, printed editions of the work, early translations, or quotations of a work in other texts. Information concerning variant readings of a text may be accumulated in highly structured form in a critical apparatus of variants. This chapter defines a module for use in encoding such an apparatus of variants, which may be used in conjunction with any of the modules defined in these Guidelines. It also defines an element class which provides extra attributes for some elements of the core tag set when this module is selected.

Information about variant readings (whether or not represented by a critical apparatus in the source text) may be recorded in a series of apparatus entries, each entry documenting one variation, or set of readings, in the text. Elements for the apparatus entry and readings, and for the documentation of the witnesses whose readings are included in the apparatus, are described in section *12.1. The Apparatus Entry, Readings, and Witnesses*. Special tags for fragmentary witnesses are described in section *12.1.5. Fragmentary Witnesses*. The available methods for embedding the apparatus in the rest of the text, or for linking an external apparatus to the base text, are described in section *12.2. Linking the Apparatus to the Text*. Finally, several extra attributes for some tags of the core tag set, made available when the additional tag set for text criticism is selected, are documented in section *11.3.1. Core elements for Transcriptional Work*.

Many examples given in this chapter refer to the following texts of the opening (usually just line 1) of Chaucer's *Wife of Bath's Prologue*, as it appears in each of the four different manuscripts

- Ellesmere, Huntingdon Library 26.C.9 (**El**)
- Hengwrt, National Library of Wales, Aberystwyth, Peniarth 392D (**Hg**)
- British Library Lansdowne 851 (**La**)
- Bodleian Library Rawlinson Poetic 149 (**Ra2**)

12.1 The Apparatus Entry, Readings, and Witnesses

This section introduces the fundamental markup methods used to encode textual variations:

- the <app> element for entries in the critical apparatus: see section *12.1.1. The Apparatus Entry*.
- elements for identifying individual readings: see section *12.1.2. Readings*.
- ways of grouping readings together: see section *12.1.3. Indicating Subvariation in Apparatus Entries*.
- methods of identifying which witnesses support a particular reading, and for describing the witnesses included in the apparatus: see section *12.1.4. Witness Information*.
- elements for indicating which portions of a text are covered by fragmentary witnesses: see section *12.1.5. Fragmentary Witnesses*.

The <app> element is in one sense a more sophisticated and complex version of the <choice> element introduced in *3.4.1. Apparent Errors* as a way of marking points where the encoding of a passage in a single source may be carried out in more than one way. Unlike <choice>, however, the <app> element allows for the representation of many different versions of the same passage taken from different sources.

12.1.1 The Apparatus Entry

Individual textual variations are encoded using the <app> element, which groups together all the readings constituting the variation. The identification of discrete textual variations or apparatus entries is not a purely mechanical process; different editors may group readings differently. No rules are given here as to how to group readings into apparatus entries; the tags given here may be used to group readings in whatever way the editor finds most perspicuous or useful.

The individual apparatus entry is encoded with the <app> element:

<app> (apparatus entry) contains one entry in a critical apparatus, with an optional lemma and at least one reading.

- @**type** classifies the variation contained in this element according to some convenient typology.
- @**from** identifies the beginning of the lemma in the base text, if necessary.
- @**to** identifies the endpoint of the lemma in the base text, if necessary.
- @**loc** (location) indicates the location of the variation, when the location-referenced method of apparatus markup is used.

The attributes **loc**, **from**, and **to**, are used to link the apparatus entry to the base text, if present. In such cases, several methods may be used for such linkage, each involving a slightly different usage for these attributes. Linkage between text and apparatus is described below in section *12.2. Linking the Apparatus to the Text*. For the use of the <app> element without a base text, see *12.2.3. The Parallel Segmentation Method*.

Each <app> element comprises one or more readings, which in turn are encoded using the <rdg> or other elements, as described in the next section. A very simple partial apparatus for the first line of the *Wife of Bath's Prologue* might take a form something like this:

```
<app>
  <rdg wit="#El">Experience though noon Auctoritee</rdg>
  <rdg wit="#La">Experiment thogh noon Auctoritee</rdg>
  <rdg wit="#Ra2">Eryment though none auctorite</rdg>
</app>
```

Of course, in practice the apparatus will be somewhat more complex. Specifically, it may be desired to record more obviously that manuscripts El and La agree on the words 'noon Auctoritee', to indicate a preference for one reading, etc. The following sections on readings, subvariation, and witness information describe some of the more important complications which can arise.

12.1.2 Readings

Individual readings are the crucial elements in any critical apparatus of variants. The following elements should be used to tag individual readings within an apparatus entry:

<lem> (lemma) contains the lemma, or base text, of a textual variation.

<rdg> (reading) contains a single reading within a textual variation.

N.B. the term lemma is used here in the text-critical sense of 'the reading accepted as that of the original or of the base text'. This sense differs from that in which the word is used elsewhere in the Guidelines, for example

as in the attribute **lemma** where the intended sense is 'the root form of an inflected word', or 'the heading of an entry in a reference book, especially a dictionary'.

In recording readings within an apparatus entry, the <rdg> element may always be used; each <app> must contain at least one <rdg>.

The <lem> element may also be used, under some circumstances, to record the base text of the source edition, to mark the readings of a base witness, to indicate the preference of an editor or encoder for a particular reading, or to make clear, in cases of ambiguity, precisely which portion of the main text the variation applies to. Those who prefer to work without the notion of a base text may prefer not to use it at all. How it is used depends in part on the method chosen for linking the apparatus to the text; for more information, see section *12.2. Linking the Apparatus to the Text*.

Readings may be encoded individually, or grouped for perspicuity using the <rdgGrp> element described in section *12.1.3. Indicating Subvariation in Apparatus Entries*.

As members of the attribute class att.textCritical, both of these elements inherit the following attributes. Some of these attributes are intelligible only if the reading is ascribed to a single witness; others have no such restriction.

att.textCritical defines a set of attributes common to all elements representing variant readings in text critical work.

- @**wit** (witness or witnesses) contains a list of one or more pointers indicating the witnesses which attest to a given reading.
- @**type** classifies the reading according to some useful typology.
- @**cause** classifies the cause for the variant reading, according to any appropriate typology of possible origins.
- @**varSeq** (variant sequence) provides a number indicating the position of this reading in a sequence, when there is reason to presume a sequence to the variants on any one lemma.
- @**hand** signifies the hand responsible for a particular reading in the witness.
- @**resp** (responsible party) identifies the editor responsible for asserting a particular reading in the witness.

The **wit** attribute identifies the witnesses which have the reading in question. It is required if the apparatus gathers together readings from different witnesses, but may be omitted in an apparatus recording the readings of only one witness, e.g. substitutions, divergent opinions on what is in the witness or on how to expand abbreviations, etc. Even in such a one-witness apparatus, however, the **wit** attribute may still be useful when it is desired to record the occurrence of a particular reading in some other witness. For other methods of identifying the witnesses to a reading, see section *12.1.4. Witness Information*.

The **type** attribute allows the encoder to classify readings in any convenient way, for example as substantive variants of the lemma:

```
<app>
  <lem wit="#El #Hg">Experience</lem>
  <rdg wit="#La" type="substantive">Experiment</rdg>
  <rdg wit="#Ra2" type="substantive">Eryment</rdg>
</app>
```

or as orthographic variants:

```
<app>
  <lem  wit="#El #Ra2">though</lem>
  <rdg  wit="#Hg"  type="orthographic">thogh</rdg>
  <rdg  wit="#La"  type="orthographic">thouh</rdg>
</app>
```

The **varSeq** and **cause** attributes may be used to convey information on the sequence and cause of variation. In the following apparatus fragment, the reading *Eryment* is tagged as sequential to (derived from) the reading *Experiment*, and the cause is given as loss of the abbreviation for *per*.

```
<app>
  <rdg  wit="#La"  varSeq="1">Experiment</rdg>
  <rdg  wit="#Ra2"  cause="abbreviation_loss"  varSeq="2">Eryment</rdg>
</app>
```

If a manuscript is written in several hands, and it is desired to report which hand wrote a particular reading, the **hand** attribute should be used. For example, in the Munich manuscript containing the *Carmina Burana*, the word *alle* has been changed to *allen*:

```
<l>Swaz hi gât umbe</l>
<l>daz sint alle megede,</l>
<l>die wellent ân man</l>
<l>
  <app>
    <rdg  wit="#Mu"  varSeq="1"  hand="#m1">alle</rdg>
    <rdg
       wit="#Mu"
       cause="nachgetragen"
       varSeq="2"
       hand="#m2">allen</rdg>
  </app>
disen sumer gân.
</l>
```

Similarly, if a witness is hard to decipher, it may be desired to indicate responsibility for the claim that a particular reading is supported by a particular witness. In line 2212a of *Beowulf*, for example, the manuscript is read in different ways by different scholars; the editor Klaeber prints one text, using parentheses to indicate his expansion, and records in the apparatus two different accounts of the manuscript reading, by Zupitza and Chambers:[1]

```
<l>se ðe on
<app>
    <rdg  wit="#Kl">hea(um) h(æþ)e</rdg>
    <rdg  wit="#ms"  resp="#Z">heaðo hlæwe</rdg>
    <rdg  wit="#ms"  resp="#Cha">heaum hope</rdg>
  </app>
</l>
<l>hord beweotode,</l>
```

[1] For the sake of legibility in the example, long marks over vowels are omitted.

Source: [14]

The **hand** and **resp** attributes are intelligible only on an element recording a reading from a single witness, and should not be used if more than one witness is given on the same <rdg> or <lem> element. If more than one witness is given for the reading, they are undefined. To convey this information when the witness is one among several, the <witDetail> element should be used; see section *12.1.4. Witness Information*.

Where there is a greater weight of editorial discussion and interpretation than can conveniently be expressed through the attributes provided on these elements (for example where there are multiple witnesses for a single reading or multiple editorial responsibility for an emendation) this information can be attached to the apparatus in a note, or recorded in the feature structure notation defined in chapter *18. Feature Structures*. In particular, such recurring text-critical situations as palaeographic confusion of particular letters, or homœoarchy or homœoteleuton involving specific character groups, may lend themselves to feature structure treatment. Information concerning these recurrent situations may be encoded into database-like fragments within the text which would then be available to sophisticated computer-assisted analysis. Further work remains to be done on such mechanisms, however, and so no examples are given here of the use of feature structures in text-critical apparatus.

The <note> element may also be used to record the specific wording of notes in the apparatus of the source edition, as here in a transcription of Friedrich Klaeber's note on *Beowulf* 2207a:

```
<l  n="2207a">syððan Beowulfe
<note  resp="#Kl"  place="app">Fol. 179a <mentioned>beowulfe</mentioned>.
    Folio 179, with the last page (Fol. 198b), is the worst part of the
    entire MS. It has been freshened up by a later hand, but not always
    correctly. Information on doubtful readings is in the notes of
    Zupitza and Chambers.</note>
</l>
<l  n="2207b">brade rice</l>
```

Source: [14]

Notes providing details of the reading of one particular witness should be encoded using the specialized <witDetail> element described in section *12.1.4. Witness Information*.

Encoders should be aware of the distinct fields of use of the attribute values **wit**, **hand**, and **resp**. Broadly, **wit** identifies the physical entity in which the reading is found (manuscript, clay tablet, papyrus, printed edition); **hand** refers to the agent responsible for inscribing that reading in that physical entity (scribe, author, inscriber, hand 1, hand 2); **resp** indicates the scholar responsible for asserting the existence of that reading in that physical entity. In some cases, the categories may blur: a scholar may produce an edition introducing readings for which he or she is responsible; that edition may itself become a witness in a later critical apparatus. Thus, readings introduced as corrections in the earlier edition will be seen in the later apparatus as witnessed by the earlier edition. As observed in the discussion concerning the discrimination of **hand** and **resp** in transcription of primary sources in section *11.4.2. Hand, Responsibility, and Certainty Attributes*, the division of layers of responsibility through various scholars for particular aspects of a particular reading may require the more complex mechanisms for assigning responsibility described in chapter *21. Certainty and Responsibility*.

12.1.3 Indicating Subvariation in Apparatus Entries

The <rdgGrp> element may be used to group readings, either because they have identical values on one or more attributes, or because they are seen as forming a self-contained variant sequence, or for some other reason. This grouping of readings is entirely optional: no such grouping of readings is required.

<rdgGrp> (reading group) within a textual variation, groups two or more readings perceived to have a genetic relationship or other affinity.

The <rdgGrp> element is a member of class att.textCritical and therefore can carry the **wit**, **type**, **cause**, **varSeq**, **hand**, and **resp** attributes described in the preceding section. When values for any of these attributes are given on a <rdgGrp> element, the values given are inherited by the <rdg> or <lem> elements nested within the reading group, unless overridden by a new specification on the individual reading element.

To indicate that both Hg and La vary only orthographically from the lemma, one might tag both readings **<rdg type='orthographic'>**, as shown in the preceding section. This fact can be expressed more perspicuously, however, by grouping their readings into a <rdgGrp>, thus:

```
<app>
  <lem  wit="#El #Ra2">though</lem>
  <rdgGrp  type="orthographic">
    <rdg  wit="#Hg">thogh</rdg>
    <rdg  wit="#La">thouhe</rdg>
  </rdgGrp>
</app>
```

Similarly, <rdgGrp> may be used to organize the substantive variants of an apparatus entry. Editors may need to indicate that each of a group of witnesses may be taken as all supporting a particular reading, even though there may be variation concerning the exact form of that reading in, or the degree of support offered by, those witnesses. For example: one may identify three substantive variants on the first word of Chaucer's *Wife of Bath's Prologue* in the manuscripts: these might be expressed in regularized spelling as *Experience*, *Experiment*, and *Eriment*. In fact, the manuscripts display many different spellings of these words, and a scholar may wish both to show that the manuscripts have all these variant spellings and that these variant spellings actually support only the three regularized spelling forms. One may term these variant spellings as 'subvariants' of the regularized spelling forms.

This subvariation can be expressed within an <app> element by gathering the readings into three groups according to the normalized form of their reading. All the readings within each group may be accounted subvariants of the main reading for the group, which may be indicated by tagging it as a <lem> element or as **<rdg type='groupBase'>**.

In this example, the different subvariants on *Experience*, *Experiment*, and *Eriment* are held within three <rdgGrp> elements nested within the enclosing <app> element:

```
<app  type="substantive">
  <rdgGrp  type="subvariants">
    <lem  wit="#El #Hg">Experience</lem>
    <rdg  wit="#Ha4">Experiens</rdg>
  </rdgGrp>
  <rdgGrp  type="subvariants">
    <lem  wit="#Cp #Ld1">Experiment</lem>
    <rdg  wit="#La">Ex<g  ref="#per"/>iment</rdg>
  </rdgGrp>
  <rdgGrp  type="subvariants">
    <lem>Eriment<wit>[unattested]</wit>
    </lem>
    <rdg  wit="#Ra2">Eryment</rdg>
  </rdgGrp>
</app>
```

From this, one may deduce that the regularized reading *Experience* is supported by all three manuscripts El Hg Ha4, although the spelling differs in Ha4, and that the regularized reading *Eriment* is supported by Ra2, even though the form differs in that manuscript. Accordingly, an application which recognizes that these apparatus

entries show subvariation may then assign all the witnesses instanced as attesting the sub-variants on that lemma as actually supporting the reading of the lemma itself at a higher level of classification. Thus, Ha4 here supports the reading *Experience* found in El and Hg, even though it is spelt slightly differently in Ha4.

Reading groups may nest recursively, so that variants can be classified to any desired depth. Because apparatus entries may also nest, the <app> element might also be used to group readings in the same way. The example above is substantially identical to the following, which uses <app> instead of <rdgGrp>:

```
<app n="a1" type="substantive">
  <rdg wit="#El #Hg #Ha4">
    <app n="a2" type="orthographic">
      <lem wit="#El #Hg">Experience</lem>
      <rdg wit="#Ha4">Experiens</rdg>
    </app>
  </rdg>
  <rdg wit="#Cp #Ld1 #La">
    <app n="a3" type="orthographic">
      <lem wit="#Cp #Ld1">Experiment</lem>
      <rdg wit="#La">Ex<g ref="#per"/>iment</rdg>
    </app>
  </rdg>
  <rdg wit="#Ra2">
    <app n="a4" type="orthographic">
      <lem>Eriment<wit>[unattested]</wit>
      </lem>
      <rdg wit="#Ra2">Eryment</rdg>
    </app>
  </rdg>
</app>
```

This expresses even more clearly than the previous encoding of this material that at the highest level of classification (apparatus entry A1), this variation has three normalized readings, and that the first of these is supported by manuscripts El, Hg, and Ha4; the second by Cp, Ld1, and La; and the third by Ra2. Some encoders may find the use of nested apparatus entries less intuitive than the use of reading groups, however, so both methods of classifying the readings of a variation are allowed.

Reading groups may also be used to bring together variants which form an apparent developmental sequence, and to make clear that other readings are not part of that sequence, as in the following example, which makes clear that the variant sequence *experiment* to *eriment* says nothing about the relative priority of *experiment* and *experience*:

```
<app type="substantive">
  <rdgGrp type="subvariants">
    <lem wit="#El #Hg">Experience</lem>
    <rdg wit="#Ha4">Experiens</rdg>
  </rdgGrp>
  <rdgGrp type="sequence">
    <rdgGrp varSeq="1" type="subvariants">
      <lem wit="#Cp #Ld1">Experiment</lem>
      <rdg wit="#La">Ex<g ref="#per"/>iment</rdg>
    </rdgGrp>
    <rdgGrp varSeq="2" cause="abbreviation_loss" resp="#PR">
      <lem>Eriment<wit>[unattested]</wit>
      </lem>
      <rdg wit="#Ra2">Eryment</rdg>
```

```
    </rdgGrp>
  </rdgGrp>
</app>
```

12.1.4 Witness Information

A given reading is associated with the set of witnesses attesting it by listing the witnesses in the **wit** attribute on the <rdg>, <lem>, or <rdgGrp> element. Special mechanisms, described in the following sections, are needed to associate annotation on a reading with one specific witness among several (section *12.1.4.1. Witness Detail Information*), to transcribe witness information verbatim from a source edition (section *12.1.4.2. Witness Information in the Source*), and to identify the formal lists of witnesses typically provided in the front matter of critical editions (section *12.1.4.3. The Witness List*).

12.1.4.1 Witness Detail Information

When it is desired to give additional information about a particular witness or witnesses for the reading, the information may be given in a <witDetail> element, pointing to the identifier for that reading and signalling in the value of its **wit** attribute the witness or witnesses to which the additional information relates.

<witDetail> (witness detail) gives further information about a particular witness, or witnesses, to a particular reading.

@**target** indicates the identifier for the reading, or readings, to which the witness detail refers.

@**wit** (witnesses) indicates the sigil or sigla for the witnesses to which the detail refers.

The <witDetail> element is a specialized form of <note>, which adds to the attributes of that element the specialized attribute **wit**, which indicates which witness in particular is being described. Like <note>, <witDetail> can be included in the text at the point of attachment, or can point to the reading(s) being annotated with its **target** attribute. To indicate, on the authority of editor PR, that the Ellesmere manuscript has an ornamental capital in the word *Experience*, for example, one might write:

```
<app type="substantive">
  <rdgGrp type="subvariants">
    <lem xml:id="W026" wit="#El #Hg">Experience</lem>
    <rdg wit="#Ha4">Experiens</rdg>
  </rdgGrp>
</app>
<witDetail target="#W026" resp="#PR" wit="#El">Ornamental capital.</witDetail>
```

This encoding makes clear that the ornamental capital mentioned is in the Ellesmere manuscript, and not in Hengwrt or Ha4.

Like <note>, <witDetail> may be used to record the specific wording of information in the source text, even when the information itself is captured in some more formal way elsewhere. The example from the Carmina Burana above (section *12.1.2. Readings*), for example, might be extended thus, to record the wording of the note explaining the variant:

```
<l>Swaz hi gât umbe</l>
<l>daz sint alle megede,</l>
<l>die wellent ân man</l>
<l>
  <app>
```

```
    <rdg  wit="#Mu"  hand="#m1">alle</rdg>
    <rdg  xml:id="anon.6.4"  wit="#Mu"  hand="#m2">allen</rdg>
  </app>
disen sumer gân.
</l>
<witDetail  target="#anon.6.4"  wit="#Mu">
  <ref>allen</ref>
  <mentioned>n</mentioned> nachgetragen.

</witDetail>
```

Observe that a single witness detail element may be linked to several different readings (noting, for example, a recurrent phenomena in a particular manuscript) by having the **target** attribute point at all the readings in question. Similarly, feature structures containing information about the text in a witness (whether retroversion, regularization, or other) can also be linked to specific <lem> and <rdg> instances. See chapter *18. Feature Structures*.

12.1.4.2 Witness Information in the Source

In the transcription of printed critical editions, it may be desirable to retain for future reference the exact form in which the source edition records the witnesses to a particular reading; this is particularly important in cases of ambiguity in the information, or uncertainty as to the correct interpretation. The <wit> element may be used to transcribe such lists of witnesses to a particular reading.

<wit> contains a list of one or more sigla of witnesses attesting a given reading, in a textual variation.

The <wit> list may appear following a <rdg>, <rdgGrp>, or <lem> element in any apparatus entry, and should be used only to transcribe the witness information in the form found in the source. The advantage of holding witness information in the **wit** attribute of <lem> or <rdg> is that an application can check that every sigil identifier has been declared elsewhere in the document. Because the **wit** attribute has declared datatype of one or more data.pointer values, a check can be made that readings are assigned only to witness sigla which have been identified (using the **xml:id** attribute) within a <listWit> element (see section *12.1.4.3. The Witness List*). Such checking is more difficult for witness sigla held as the content of a <wit> element. For this reason, it is recommended that encoders always hold witness information in the **wit** attribute of <lem> and <rdg>, where possible. Thus, as in the examples below, even when a reference to a witness is exactly reproduced in the <wit> element, the corresponding sigil for that witness can be written into the **wit** attribute of the matching <rdg> or <lem>. However, in cases where it is uncertain how the witness reference contained in the <wit> element should be interpreted, or where no witness exists, the **wit** attribute on the matching <rdg> or <lem> may be left empty.

```
<lg  type="stanza">
  <l  xml:id="Diet1.1">Slăfest du, vriedel ziere?</l>
  <l  xml:id="Diet1.2">wan wecket uns leider schiere;</l>
  <l  xml:id="Diet1.3">ein vogellīn sŏ wol getăn</l>
  <l  xml:id="Diet1.4">daz ist der linden an daz zwī gegăn.</l>
</lg>
<app  type="secondary"  loc="Diet.1.1">
  <rdg  wit="#Kb">slăfst</rdg>
  <wit>K(Ba)</wit>
</app>
<app  type="secondary"  loc="Diet.1.2">
  <rdg  wit="#Kv">Man</rdg>
  <wit>K(V)</wit>
```

```
  <rdg  wit="#K">weckt</rdg>
  <wit>K (Wackernagel 401)</wit>
  <rdg  wit="#Ju">Ich waen ez taget uns schiere</rdg>
  <wit>Jungbluth, Festschr. Pretzel 1963, 122.</wit>
</app>
```

Of course, the siglum used for a particular witness in the source, as recorded in the <wit> element, may well differ from that used to indicated the same witness in the **wit** attribute, as shown particularly in the apparatus for the second line of the poem (Diet.1.2).

12.1.4.3 The Witness List

A list of all identified witnesses should normally be supplied in the front matter of the edition, or in the <sourceDesc> element of its header. This may be given either as a simple bibliographic list, using the <listBibl> element described in *3.11. Bibliographic Citations and References*, or as a <listWit> element, which contains a series of <witness> elements. Each <witness> element may contain a brief characterisation of the witness, given as one or more prose paragraphs. If more detailed information about a manuscript witness is available, it should be represented using the <msDesc> element provided by the msdescription module; a <msDesc> may appear within a <listBibl>.

Whether information about a particular witness is supplied by means of a <bibl>, <msDesc>, or <witness> element, a unique sigil (siglum) for this source should always be supplied, using the global **xml:id** attribute. This identifier can then be used elsewhere to refer to this particular witness.

<listWit> (witness list) lists definitions for all the witnesses referred to by a critical apparatus, optionally grouped hierarchically.

<witness> contains either a description of a single witness referred to within the critical apparatus, or a list of witnesses which is to be referred to by a single sigil.

<msDesc> (manuscript description) contains a description of a single identifiable manuscript.

<bibl> (bibliographic citation) contains a loosely-structured bibliographic citation of which the sub-components may or may not be explicitly tagged.

<listBibl> (citation list) contains a list of bibliographic citations of any kind.

The minimal information provided by a witness list is thus the set of sigla for all the witnesses named in the apparatus. For example, the witnesses referenced by the examples of this chapter might simply be listed thus:

```
<listWit>
  <witness  xml:id="Chi3"/>
  <witness  xml:id="Ha4"/>
  <witness  xml:id="Ju"/>
  <witness  xml:id="K"/>
  <witness  xml:id="Kl"/>
  <witness  xml:id="Ld1"/>
  <witness  xml:id="Mu"/>
  <witness  xml:id="Kb"/>
  <witness  xml:id="Kv"/>
  <witness  xml:id="Wa"/>
  <witness  xml:id="X"/>
</listWit>
```

It is more helpful, however, for witness lists to be somewhat more informative: each <witness> element should contain at least a brief prose description of the witness, perhaps including a bibliographic citation, as in the following examples:

```
<listWit>
  <witness  xml:id="El">Ellesmere, Huntingdon Library 26.C.9</witness>
  <witness  xml:id="Hg">Hengwrt, National Library of Wales,
     Aberystwyth, Peniarth 392D</witness>
  <witness  xml:id="Ra2">Bodleian Library Rawlinson Poetic 149
     (see further <ptr  target="#MSRP149"/>)</witness>
</listWit>
```

As the last example shows, the witness description here may be complemented by a reference to a full description of the manuscript supplied elsewhere, typically as the content of a <msDesc> or <bibl> element. Alternatively, it may contain a whole paragraph of commentary for each witness :

```
<listWit>
  <witness  xml:id="A">die sog. <soCalled>Kleine (oder alte)
        Heidelberger Liederhandschrift</soCalled>.
  <bibl>Universitätsbibliothek Heidelberg col. pal.
        germ. 357. Pergament, 45 Fll. 18,5 × 13,5 cm.</bibl>
     Wahrscheinlich die älteste der drei großen Hss. Sie
  <quote>datiert aus dem 123. Jahrhundert, etwa um 1275. Ihre Sprache
        weist ins Elsaß, evtl. nach Straßburg. Man geht wohl nicht
        fehl, in ihr eine Sammlung aus dem Stadtpatriziat zu sehen</quote>
     (<bibl>
      <author>Blank</author>, [vgl. <ref>Lit. z. Hss. Bd. 2,
           S. 39</ref>] S. 14</bibl>). Sie enthält 34 namentlich
     genannte Dichter. <quote>Zu den Vorzügen von A gehört, daß
        sie kaum je bewußt geändert hat, so daß sie für
        manche Dichter ... oft den besten Text liefert</quote> (so wohl mit
     Recht <bibl>
      <author>v. Kraus</author>
    </bibl>).</witness>
  <witness  xml:id="a">Bezeichnung <bibl>
      <author>Lachmann</author>
    </bibl>s für die von einer 2. Hand auf bl. 40–43
     geschriebenen Strophen der Hs. A.</witness>
  <witness  xml:id="B">die <soCalled>Weingartner (Stuttgarter)
        Liederhandschrift</soCalled>. <bibl>Württembergische
        Landesbibliothek Stuttgart, HB XIII poetae germanici 1.
        Pergament, 156 Bll. 15 × 11,5 cm; 25 teils ganzseitig,
        teils halbseitige Miniaturen.</bibl> Kaum vor 1306 in Konstanz
     geschrieben. Sie enthält Lieder von 25 namentlich genannten
     Dichtern. (Dazu kommen Gedichte von einigen ungenannten
     bzw. unbekannten Dichtern, ein Marienlobpreis und eine
     Minnelehre.)</witness>
</listWit>
```

It would however generally be preferable to represent such detailed information using an appropriately structured <msDesc> element, as discussed in chapter *10. Manuscript Description.* Note also that if the witnesses being recorded are not manuscripts but printed works, it may be preferable to document them using the standard <bibl> or <biblStruct> elements described in *3.11. Bibliographic Citations and References*, as in this example:

```
<listBibl>
  <bibl  xml:id="bcn_1482">T.Kempis, De la imitació de Jesuchrist e del
```

```
    menyspreu del món (trad. Miquel Peres); Barcelona, 1482, Pere
    Posa. Editio princeps.</bibl>
  <bibl xml:id="val_1491">T.Kempis, Del menyspreu del món (trad. Miquel
    Peres); València, 1491.</bibl>
  <bibl xml:id="bcn_1518">T.Kempis, Libre del menysprey del món e de la
    imitació de nostre senyor Déu Jesucrist, (trad. Miquel Peres);
    Barcelona, 1518, Carles Amorós. </bibl>
</listBibl>
```

In text-critical work it is customary to refer to frequently occurring groups of witnesses by means of a single common sigil. Such sigla may be documented as pseudo-witnesses in their own right by including a nested witness list within the witness list, which uses the sigil for the group as its identifier, and supplies a fuller name for the group in its optional child <head> element, before listing the other witnesses contained by the group. For example, the Constant Group C of manuscripts comprising witnesses Cp, La, and S12, might be represented as follows:

```
<listWit>
  <witness xml:id="Ellesmere">Ellesmere, Huntingdon Library 26.C.9</witness>
<!-- ... -->
  <listWit xml:id="Con">
    <head>Constant Group C</head>
    <witness xml:id="Cp">Corpus Christi Oxford MS 198</witness>
    <witness xml:id="La">British Library Lansdowne 851</witness>
    <witness xml:id="Sl2">British Library Sloane MS 1686</witness>
  </listWit>
</listWit>
```

That the reading *Experiment* occurs in all three manuscripts can now be indicated simply as follows:

```
<rdg wit="#Con">Experiment</rdg>
```

Note that a single witness cannot appear more than once in a witness list, and therefore cannot be assigned to more than one group of witnesses.

Situations commonly arise where there are many more or less fragmentary witnesses, such that there may be quite distinct groups of witnesses for different parts of a text or collection of texts. One may treat this with distinct <listWit> elements for each different part. Alternatively, one may have a single <listWit> element at the beginning of the file or in its header listing all the witnesses, partial and complete, for the text, with the attestation of fragmentary witnesses indicated within the apparatus by use of the <witStart> and <witEnd> elements described in section *12.1.5. Fragmentary Witnesses*.

If a witness list is provided, it may be unnecessary to give, in each apparatus entry, an exhaustive list of the witnesses which agree with the base text. An application program can — in principle — compare the witnesses given for each variant found with those given in the full list of witnesses, subtracting from this list all the witnesses not active at this point (perhaps because of lacuna, or because they contain a variation on a different, overlapping lemma) and thence calculate all the manuscripts agreeing with the base text. In practice, encoders may find it less error-prone to list all witnesses explicitly in each apparatus entry.

12.1.5 Fragmentary Witnesses

If a witness is incomplete (whether a single fragment, a series of fragments, or a relatively complete text with one or more lacunae), it is usually desirable to record explicitly where its preserved portions begin and end.

The following empty tags, which may occur within any <lem> or <rdg> element, indicate the beginning or end of a fragmentary witness or of a lacuna within a witness:

<witStart/> (fragmented witness start) indicates the beginning, or resumption, of the text of a fragmentary witness.

<witEnd/> (fragmented witness end) indicates the end, or suspension, of the text of a fragmentary witness.

<lacunaStart/> indicates the beginning of a lacuna in the text of a mostly complete textual witness.

<lacunaEnd/> indicates the end of a lacuna in a mostly complete textual witness.

These elements constitute the class model.rdgPart, members of which are permitted within the elements <lem> and <rdg> when the module defined by this chapter is included in a schema.

Suppose a fragment of a manuscript X of the *Wife of Bath's Prologue* has a physical lacuna, and the text of the manuscript begins with *auctorite*. In an apparatus this might appear thus, distinguished from the reading of other manuscripts by the presence of the <lacunaEnd> element:

```
<app>
  <lem  wit="#El #Hg">Auctoritee</lem>
  <rdg  wit="#La #Ra2">auctorite</rdg>
  <rdg  wit="#X">
    <lacunaEnd/>auctorite</rdg>
</app>
```

In some cases, the apparatus in the source may commence recording the readings for a particular witness without its being clear whether the previous absence of readings for this witness is due to a lacuna, or to some other reason. The <witStart> element may be used in this circumstance:

```
<app>
  <lem  wit="#El #Hg">Auctoritee</lem>
  <rdg  wit="#La #Ra2">auctorite</rdg>
  <rdg  wit="#X">
    <witStart/>auctorite</rdg>
</app>
```

12.2 Linking the Apparatus to the Text

Three different methods may be used to link a critical apparatus to the text:

1. the location-referenced method,
2. the double-end-point-attached method, and
3. the parallel segmentation method.

Both the location-referenced and the double end-point methods may be used with either in-line or external apparatus, the former dispersed within the base text, the latter held in some separate location, within or outside the document with the base text. The parallel segmentation method does not use the concept of a base text and may only be used for in-line apparatus.

Any document containing <app> elements requires a <variantEncoding> declaration in the <encodingDesc> element of its TEI header, thus:

<variantEncoding/> declares the method used to encode text-critical variants.

@**method** indicates which method is used to encode the apparatus of variants.

@**location** indicates whether the apparatus appears within the running text or external to it.

12.2.1 The Location-referenced Method

The location-referenced method of encoding apparatus provides a convenient method for encoding printed apparatus; in this method as in most printed editions, the apparatus is linked to the base text by indicating explicitly only the block of text on which there is a variant (noted usually by a canonical reference scheme, or by line number in the edition, such as **A 137** or *Page 15 line 1*).

If the location-referenced method is used for an apparatus stored externally to the base text, the TEI header must have the declaration:

```
<variantEncoding method="location-referenced" location="external"/>
```

In the <body> of the document, the base text (here El) will appear:

```
<text>
  <body>
    <div n="WBP" type="prologue">
      <head>The Prologe of the Wyves Tale of Bathe</head>
      <l n="1">Experience though noon Auctoritee</l>
      <l>Were in this world ...</l>
    </div>
  </body>
</text>
```

Elsewhere in the document, or in a separate file, the apparatus will appear. On each <app> element, the **loc** attribute should be specified to indicate where the variant occurs in the base text.

```
<app loc="WBP 1">
  <rdg wit="#La">Experiment</rdg>
  <rdg wit="#Ra2">Eryment</rdg>
</app>
```

If the same text is encoded using in-line storage, the apparatus is dispersed through the base text block to which it refers. In this case, the location of the variant can be read from the line in which it occurs.

```
<variantEncoding method="location-referenced" location="internal"/>
<!-- ... -->
<l n="1">Experience
<app>
    <rdg wit="#La">Experiment</rdg>
    <rdg wit="#Ra2">Eryment</rdg>
  </app>
though noon Auctoritee</l>
<l>Were in this world ...</l>
```

Since the location is not required to be exact, the apparatus for a line might also appear at the end of the line:

```
<l n="1">Experience though noon Auctoritee
<app>
    <rdg wit="#La"> Experiment</rdg>
```

```
    <rdg  wit="#Ra2"> Eryment</rdg>
  </app>
</l>
<l>Were in this world ...</l>
```

When the apparatus is linked to the text by means of location references, as shown here, it is not possible to find automatically the precise portion of text varied by the readings. In order to show explicitly what portion of the base text is replaced by the variant readings, the <lem> element may be used:

```
<l  n="1">Experience though noon Auctoritee
<app>
    <lem  wit="#El">Experience</lem>
    <rdg  wit="#La">Experiment</rdg>
    <rdg  wit="#Ra2">Eryment</rdg>
  </app>
</l>
<l>Were in this world ...</l>
```

Often the lemma will have no attributes, being simply the 'base text reading' and requiring no qualification, but it may optionally carry the normal attributes, as shown here. Some text critics prefer to abbreviate or elide the lemma, in order to save space or trouble; such practice is not forbidden by these Guidelines, but no recommendations are made for conventions of abbreviating the lemma, whether abbreviation of each word, or suppression of all but the first and last word, etc.

Where it is intended that the apparatus be complete enough to allow the reconstruction of the witnesses (or at least of their non-orthographic variations), simple location-reference methods are unlikely to be as successful as the other two methods, which allow the unambiguous reconstruction of the lemma from the encoding. The use of (for example) an XPath expression denoting a text range rather than a simple pointer may however obviate this necessity

12.2.2 The Double End-Point Attachment Method

In the double end-point attachment method, the beginning and end of the lemma in the base text are both explicitly indicated. It thus differs from the location-referenced method, in which only the larger span of text containing the lemma is indicated. Double end-point attachment permits unambiguous matching of each variant reading against its lemma. It or the parallel-segmentation method should be used in all cases where this is desired, for example where the apparatus is intended to enable full reconstruction of the text, or of the substantives, of every witness.

When the double end-point attachment method is used, the **from** and **to** attributes of the <app> element are used to indicate the beginning and ending points of the reading in the base text: their values are identifiers which occur at the locations in question. If no other markup is present there, the beginning and ending points should be marked using the <anchor> element defined in chapter *16. Linking, Segmentation, and Alignment*. In cases where it is not possible to insert anchors within the base text (e.g. where the text is on a read-only medium) the beginning and end of the lemma may be indicated by using the 'indirect pointing' mechanisms discussed in chapter *16. Linking, Segmentation, and Alignment*. Explicit anchors are more likely to be reliable, and are therefore to be preferred.

The double end-point attachment method may be used with in-line or external apparatus. In the latter case, the base text (here El) will appear with <anchor> elements inserted at every place where a variant begins or ends (unless some element with an identifier already begins or ends at that point):

```
<variantEncoding method="double-end-point" location="external"/>
<!-- ... -->
<div n="WBP" type="prologue">
  <head>The Prologe ... </head>
  <l n="1" xml:id="WBP.1">Experience<anchor xml:id="WBP-A2"/> though noon Auctoritee</l>
  <l>Were in this world ...</l>
</div>
```

The apparatus will be separately encoded:

```
<app from="#WBP.1" to="#WBP-A2">
  <rdg wit="#La">Experiment</rdg>
  <rdg wit="#Ra2">Eryment</rdg>
</app>
```

No <anchor> element is needed at the beginning of the line, since the **from** attribute can use the identifier for the line as a whole; the lemma is assumed to run from the beginning of the element indicated by the **from** attribute, to the end of that indicated by the **to** attribute. If no value is given for **to**, the lemma runs from the beginning to the end of the element indicated by the **from** attribute.

When the apparatus is encoded in-line, it is dispersed through the base text. Only the beginning of the lemma need be marked with an <anchor>, since the <app> is inserted at the end of the lemma, and itself therefore marks the end of the lemma.

```
<variantEncoding method="double-end-point" location="internal"/>
<!-- ... -->
<l n="1" xml:id="wbp.1">Experience
<app from="#wbp.1">
   <rdg wit="#La">Experiment</rdg>
   <rdg wit="#Ra2">Eryment</rdg>
  </app>
though noon Auctoritee</l>
<l>Were in this world ...</l>
```

The lemma need not be repeated within the <app> element in this method, as it may be extracted reliably from the base text. If an exhaustive list of witnesses is available, it will also not be necessary to specify just which manuscripts agree with the base text to enable reconstruction of witnesses. An application will be able to determine the manuscripts that witness the base reading, by noting which witnesses are attested as having a variant reading, and inferring the base text reading for all others after adjusting for fragmentary witnesses and for witnesses carrying overlapping variant readings.

Alternatively, if it is desired to make an explicit record of the attestation of the base text, the <lem> element may be embedded within <app>, carrying the witnesses to the base. Thus

```
<app from="#WBP.1" to="#WBP-A2">
  <lem wit="#El #Hg">Experience</lem>
  <rdg wit="#La">Experiment</rdg>
  <rdg wit="#Ra2">Eryment</rdg>
</app>
```

This method is designed to cope with 'overlapping lemmata'. For example, at line 117 of the Wife of Bath's Prologue, the manuscripts Hg (Hengwrt), El (Ellesmere), and Ha4 (British Library Harleian 7334) read:

Hg And of so parfit wys a wight ywroght

El And for what profit was a wight ywroght

Ha4 And in what wise was a wight ywroght

In this case, one might wish to record *in what wise was* in Ha4 as a single variant for *of so parfit wys* in Hg, and *was a wight* in El and Ha4 as a variant on *wys a wight* in Hg. This method can readily cope with such difficult situations, typically found in large and complex traditions:

```
<l xml:id="WBP.117" n="117"> And
<anchor xml:id="WBP-A117.1"/> of so parfit
<anchor xml:id="WBP-A117.2"/> wys
<anchor xml:id="WBP-A117.3"/> a wight
<anchor xml:id="WBP-A117.4"/> ywroght
<app from="#WBP-A117.1" to="#WBP-A117.3">
   <lem wit="#Hg">of so parfit wys</lem>
   <rdg wit="#Ha4">in what wise was</rdg>
 </app>
 <app from="#WBP-A117.2" to="#WBP-A117.4">
   <lem wit="#Hg">wys a wight</lem>
   <rdg wit="#El #Ha4">was a wight</rdg>
 </app>
</l>
```

The parallel segmentation method, to be discussed next, cannot handle overlaps among variants, and would require the individual variants to be split into pieces.

Because creation and interpretation of double end-point attachment apparatus will be lengthy and difficult it is likely that they will usually be created and examined by scholars only with mechanical assistance.

12.2.3 The Parallel Segmentation Method

This method differs from the double end-point attachment method in that all variants at any point of the text are expressed as variants on one another. In this method, no two variations can overlap, although they may nest. Thus, the concepts of a base text and of a lemma become unnecessary: the texts compared are divided into matching segments all synchronized with one another. This permits direct comparison of any span of text in any witness with that in any other witness. It is also very easy with this method for an application to extract the full text of any one witness from the apparatus.

This method will (by definition) always be satisfactory when there are just two texts for comparison (assuming they are in the same language and script). It will also be useful where editors do not wish to privilege a text as the 'base' or when editors wish to present parallel texts. It will become less convenient as traditions become more complex and tension develops between the need to segment on the largest variation found and the need to express the finest detail of agreement between witnesses.

In the parallel segmentation method, each segment of text on which there is variation is marked by an <app> element; each reading is given in a <rdg> element; if it is desired to single out one reading as preferred, it may be tagged <lem>:

```
<variantEncoding method="parallel-segmentation" location="internal"/>
<!-- ... -->
<l n="1">
  <app>
    <lem wit="#El #Hg">Experience</lem>
```

```
   <rdg wit="#La">Experiment</rdg>
   <rdg wit="#Ra2">Eryment</rdg>
  </app>
though noon Auctoritee
</l>
<l>Were in this world ...</l>
```

This method cannot be used with external apparatus: it must be used in-line. Note that apparatus encoded with this method may be translated into the double end-point attachment method and back without loss of information. Where double-end-point-attachment encodings have no overlapping lemmata, translation of these to the parallel segmentation encoding and back will also be possible without loss of information.

For economy, the witnesses to the reading most widely attested need not be stated. Since all manuscripts must be represented in all apparatus entries, it will be possible for an application to read a <listWit> declaring all the witnesses to the text and then calculate which witnesses have not been named. In the example below, only La and Ra2 are identified explicitly with a reading; an application might successfully infer from this that *Experience*, whose witnesses are not given, must be attested by El and Hg. To avoid confusion, however, witnesses may be omitted only for a single reading.

```
<l n="1">
  <app>
    <lem>Experience</lem>
    <rdg wit="#La">Experiment</rdg>
    <rdg wit="#Ra2">Eryment</rdg>
  </app>
though noon Auctoritee
</l>
<l>Were in this world ...</l>
```

Alternatively, the witnesses for every reading may be stated, as in the first example.

As noted, apparatus entries may nest in this method: if an imaginary fifth manuscript of the text read *Auctoritee, though none experience*, the variation on the individual words of the line would nest within that for the line as a whole:

```
<l n="1">
  <app>
    <rdg wit="#Chi3">Auctoritee, though none experience</rdg>
    <rdg>
      <app>
        <rdg wit="#El #Hg">Experience</rdg>
        <rdg wit="#La">Experiment</rdg>
        <rdg wit="#Ra2">Eryment</rdg>
      </app>
      <app>
        <rdg wit="#El #Ra2">though</rdg>
        <rdg wit="#Hg">thogh</rdg>
        <rdg wit="#La">thouh</rdg>
      </app>
      <app>
        <rdg wit="#El #Hg">noon Auctorite</rdg>
        <rdg wit="#La #Ra2">none auctorite</rdg>
      </app>
    </rdg>
```

```
  </app>
</l>
```

Parallel segmentation cannot, however, deal very gracefully with variants which overlap without nesting: such variants must be broken up into pieces in order to keep all witnesses synchronized.

12.3 Using Apparatus Elements in Transcriptions

It is often desirable to record different transcriptions of one stretch of text. These variant transcriptions may be grouped within a single <app> element. An application may then construct different 'views' of the transcription by extraction of the appropriate variant readings from the apparatus elements embedded in the transcription.

For example, alternative expansions can be recorded in several different <expan> elements, all grouped within an <app> element. Consider, for example, the three different transcriptions given below of line 105 of the Hengwrt manuscript of Chaucer's *The Wife of Bath's Prologue*. The last word of the line *Virginite is grete perfection* is written *perfectio* followed by two minims over which a bar has been drawn, which has been read in different ways by different scholars. The first transcription, by Elizabeth Solopova, represents the two minims with bar above as a special composite character using the <g>element. This transcription notes this as a mark of abbreviation but gives no expansion for it. A second transcriber, F. J. Furnivall, regards the bar as an abbreviation of *u*, and therefore reads the two minims as an *n*. A third transcriber, P. G. Ruggiers, regards the bar as an abbreviation of *n*, reading the minims as *u*. This information may be held within an <app> structure, as follows:

```
Virginite is grete
<app>
  <rdg  resp="#ES">perfectio<am>
     <g  ref="#ii"/>
    </am>
  </rdg>
  <rdg  resp="#FJF">perfectio<ex>u</ex>n</rdg>
  <rdg  resp="#PGR">perfectiou<ex>n</ex>
  </rdg>
</app>
```

This example uses special purpose elements <am> and <ex> used to represent abbreviation marks and editorial expansion respectively; these elements are provided by the transcr module documented in chapter *11. Representation of Primary Sources*, which should be consulted for further discussion of methods of representing multiple readings of a source.

Editorial notes may also be attached to <app> structures within transcriptions. Here, editorial preference for Ruggiers' expansion and an explanation of that preference is given:

```
Virginite is grete
<app>
  <rdg  resp="#ES">perfecti<am>
     <g  ref="#ii"/>
    </am>
  </rdg>
  <rdg  xml:id="f105"  resp="#FJF">perfectio<ex>u</ex>n</rdg>
  <rdg  xml:id="r105"  resp="#PGR">perfectiou<ex>n</ex>
  </rdg>
</app>
```

```
<!-- ... <note> appearing elsewhere in the document ... -->
<note  target="#r105 #f105">Furnivall's expansion implies that the bar
is an abbreviation for 'u'. There are no certain instances of
this mark as an abbreviation for 'u' in these manuscripts and it is
widely used as an abbreviation for 'n'. Ruggiers' expansion is to
be accepted.</note>
```

In most cases, elements used to indicate features of a primary textual source may be represented within an <app> structure simply by nesting them within its readings, just as the <abbr> and <expan> elements are nested within the <rdg> elements in the example just given. However, in cases where the tagged feature extends across a span of text which might itself contain variant readings which it is desired to represent by <app> structures, some adaptation of the tagging may be necessary. For example, a span of text may be marked in the transcription of the primary source as a single deletion but it may be desirable to represent just a few words from this source as individual deletions within the context of a critical apparatus drawing together readings from this and several other witnesses. In this case, the tagging of the span of words as one deletion may need to be decomposed into a series of one-word deletions for encoding within the apparatus. If it is important to record the fact that all were deleted by the same act, the markup may use the <join> element or the **next** and **prev** attributes defined by chapter *16. Linking, Segmentation, and Alignment*.

12.4 Module for Critical Apparatus

The module described in this chapter makes available the following components:

Module textcrit: Critical Apparatus

- Elements defined: app lacunaEnd lacunaStart lem listWit rdg rdgGrp variantEncoding wit witDetail witEnd witStart witness
- Classes defined: att.rdgPart att.textCritical model.rdgLike model.rdgPart

The selection and combination of modules to form a TEI schema is described in *1.2. Defining a TEI Schema*.

Chapter 13

Names, Dates, People, and Places

This chapter describes a module which may be used for the encoding of names and other phrases descriptive of persons, places, or organizations, in a manner more detailed than that possible using the elements already provided for these purposes in the Core module. In section *3.5. Names, Numbers, Dates, Abbreviations, and Addresses* it was noted that the elements provided in the core module allow an encoder to specify that a given text segment is a proper noun, or a referring string, and to specify the kind of object named or referred to only by supplying a value for the **type** attribute. The elements provided by the present module allow the encoder to supply a detailed sub-structure for such referring strings, and to distinguish explicitly between names of persons, places, and organizations.

This module also provides elements for the representation of information about the person, place, or organization to which a given name is understood to refer and to represent the name itself, independently of its application. In simple terms, where the core module allows one simply to represent that a given piece of text is a name, this module allows one further to represent a personal name, to represent the person being named, and to represent the canonical name being used. A similar range is provided for names of places and organizations. The main intended applications for this module are in biographical, historical, or geographical data systems such as gazetteers and biographical databases, where these are to be integrated with encoded texts.

The chapter begins by discussing attributes common to many of the elements discussed in the remaining parts of the chapter (*13.1. Attribute Classes Defined by this Module*) before discussing specifically the elements provided for the encoding of component parts of personal names (section *13.2.1. Personal Names*), place names (section *13.2.3. Place Names*) and organizational names (section *13.2.2. Organizational Names*). Elements for encoding personal and organizational data are discussed in section *13.3. Biographical and Prosopographical Data*. Elements for the encoding of geographical data are discussed in section *13.3.4. Places*. Finally, elements for encoding onomastic data are discussed in *13.3.5. Names and Nyms*, and the detailed encoding of dates and times is described in section *13.3.6. Dates and Times*.

13.1 Attribute Classes Defined by this Module

Most of the elements made available by this chapter share some important characteristics which are expressed by their membership in specific attribute classes. Members of the class att.naming have specialized attributes which support linkage of a naming element with the entity (person, place, organization) being named; members of the class att.datable have specialized attributes which support a number of ways of normalizing the date or time of the data encoded by the element concerned.

13.1.1 Linking Names and their Referents

As members of the att.naming class, many of the elements described in this chapter share the following attributes:

att.naming provides attributes common to elements which refer to named persons, places, organizations etc.

- @**key** provides an external means of locating a full definition for the entity (or entities) being named, such as a database record key or other token.
- @**nymRef** (reference to the canonical name) provides a means of locating the canonical form (nym) of the names associated with the object named by the element bearing it.
- @**ref** (reference) provides an explicit means of locating a full definition for the entity being named by means of a URI.

These attributes provide two different ways of associating any sort of name with its referent, and also provide a way of associating a name with its canonical form. The encoder may use these attributes in combination as appropriate. The **ref** attribute should be used wherever it is possible to supply a direct link such as a URI to indicate the location of canonical information about the referent. For example:

```
That silly man
<name  ref="#DPB1"  type="person">David Paul Brown</name> has suffered ...
```

This encoding requires that there exist somewhere a <person> element with the identifier `DPB1`, which will contain canonical information about this particular person, marked up using the elements discussed in *13.3. Biographical and Prosopographical Data* below. The same element might alternatively be provided by some other document, of course, which the same attribute could refer to by means of a URI, as explained in *16.2. Pointing Mechanisms*:

```
That silly man
<name
   ref="http://www.example.com/personography.xml#DPB1"
   type="person">David Paul Brown</name> has suffered
...
```

More than one URI may be supplied if the name refers to more than one person. For example, assuming the existence of another <person> element for Mrs Brown, with identifier `EBB1`, a reference to 'the Browns' might be encoded

```
That wretched pair
<name  ref="#DPB1 #EBB1"  type="person">the Browns</name> came to dine
...
```

The **key** attribute is provided for cases where no such direct link is required: for example because resolution of the reference is carried out by some local convention, or because the encoder judges that no such resolution is necessary. As an example of the first case, a project might maintain its own local database system containing canonical information about persons and places, each entry in which is accessed by means of some system-specific identifier constructed in a project-specific way from the value supplied for the **key** attribute.[1] As an example of the second case, consider the use of well-established codifications such as country or airport codes, which it is probably unnecessary for an encoder to expand further:

[1] In the module described by chapter *22. Documentation Elements* a similar method is used to link element descriptions to the modules or classes to which they belong, for example.

```
I never fly from <name  key="LHR"  type="place">Heathrow Airport</name>
to
<name  key="FR"  type="place">France</name>
```

The **nymRef** attribute has a more specialised use, where it is the name itself which is of interest rather than the person, place, or organization being named. See section *13.3.5. Names and Nyms* for further discussion.

Some members of the att.naming class are also members of the att.editLike class, from which they inherit the following attributes:

att.editLike provides attributes describing the nature of a encoded scholarly intervention or interpretation of any kind.

- @**resp** (responsible party) indicates the agency responsible for the intervention or interpretation, for example an editor or transcriber.
- @**cert** (certainty) signifies the degree of certainty associated with the intervention or interpretation.

This enables an encoder to record the agency responsible for a given assertion (for example, the name) and the confidence placed in that assertion by the encoder.

Examples are given below.

13.1.2 Dating Attributes

Members of the att.datable class share the following attributes:

att.datable.w3c provides attributes for normalization of elements that contain datable events using the W3C datatypes.

- @**when** supplies the value of a date or time in a standard form, e.g. yyyy-mm-dd.
- @**notBefore** specifies the earliest possible date for the event in standard form, e.g. yyyy-mm-dd.
- @**notAfter** specifies the latest possible date for the event in standard form, e.g. yyyy-mm-dd.
- @**from** indicates the starting point of the period in standard form, e.g. yyyy-mm-dd.
- @**to** indicates the ending point of the period in standard form, e.g. yyyy-mm-dd.

These attributes can be combined in various ways to provide additional nuances. Consider, for example, the following:

```
<birth  when="1857-03-15">15 March 1857.</birth>
```

```
<birth  notBefore="1857-03-01"  notAfter="1857-04-30">Some time
in March or April of 1857.</birth>
```

```
<residence  from="1857-03-01"  to="1857-04-30">In March and April of 1857.</residence>
```

```
<residence  from="1857-03-01"  notAfter="1857-04-30">From the 1st of March to
some time in April of 1857.</residence>
```

More specialized standard date formats may also be used, about which see *13.3.6.3. More Expressive Normalizations*.

13.2 Names

13.2.1 Personal Names

The core <rs> and <name> elements can distinguish names in a text but are insufficiently powerful to mark their internal components or structure. To conduct nominal record linkage or even to create an alphabetically sorted list of personal names, it is important to distinguish between a family name, a forename and an honorary title. Similarly, when confronted with a referencing string such as 'John, by the grace of God, king of England, lord of Ireland, duke of Normandy and Aquitaine, and count of Anjou', the analyst will often wish to distinguish amongst the various constituent elements present, since they provide additional information about the status, occupation, or residence of the person to whom the name belongs. The following elements are provided for these and related purposes:

<persName> (personal name) contains a proper noun or proper-noun phrase referring to a person, possibly including any or all of the person's forenames, surnames, honorifics, added names, etc.

<surname> contains a family (inherited) name, as opposed to a given, baptismal, or nick name.

<forename> contains a forename, given or baptismal name.

<roleName> contains a name component which indicates that the referent has a particular role or position in society, such as an official title or rank.

<addName> (additional name) contains an additional name component, such as a nickname, epithet, or alias, or any other descriptive phrase used within a personal name.

<nameLink> contains a connecting phrase or link used within a name but not regarded as part of it, such as *van der* or *of*.

<genName> (generational name component) contains a name component used to distinguish otherwise similar names on the basis of the relative ages or generations of the persons named.

In addition to the att.naming attributes mentioned above, all of the above elements are members of the class att.personal, and thus share the following attributes:

att.personal (attributes for components of personal names) common attributes for those elements which form part of a personal name.

- @**full** indicates whether the name component is given in full, as an abbreviation or simply as an initial.
- @**sort** specifies the sort order of the name component in relation to others within the personal name.

The <persName> element may be used in preference to the general <name> element irrespective of whether or not the components of the personal name are also to be marked. The element <persName> is synonymous with the element **<name type="person">**, except that its **type** attribute allows for further subcategorization of the personal name itself, for example as a **married**, **maiden**, **pen**, **pseudo**, or **religious** name. Consequently the following examples are equivalent:

```
That silly man
<rs  key="DPB1"  type="person">David Paul Brown</rs> has suffered the
furniture of his office to be seized
the third time for rent.
```

```
That silly man
<rs  key="DPB1"  type="person">
  <name>David Paul Brown</name>
</rs> has suffered ...
```

```
That silly man
<name  key="DPB1"  type="person">David Paul Brown</name> has suffered ...
```

```
That silly man
<persName  key="DPB1">David Paul Brown</persName> has suffered ...
```

The <persName> element is more powerful than the <rs> and <name> elements because distinctive name components occurring within it can be marked as such.

Many cultures distinguish between a family or inherited surname and additional personal names, often known as given names. These should be tagged using the <surname> and <forename> elements respectively and may occur in any order:

```
<persName>
  <surname>Roosevelt</surname>,
<forename>Franklin</forename>
  <forename>Delano</forename>
</persName>
<persName>
  <forename>Franklin</forename>
  <forename>Delano</forename>
  <surname>Roosevelt</surname>
</persName>
```

The **type** attribute may be used with both <forename> and <surname> elements to provide further culture- or project- specific detail about the name component, for example:

```
<persName>
  <forename  type="first">Franklin</forename>
  <forename  type="middle">Delano</forename>
  <surname>Roosevelt</surname>
</persName>
<persName>
  <forename  type="given">Margaret</forename>
  <forename  type="unused">Hilda</forename>
  <surname  type="maiden">Roberts</surname>
  <surname  type="married">Thatcher</surname>
</persName>
<persName  type="religious">Muhammad Ali</persName>
<persName>
  <forename>Norman</forename>
  <surname  type="complex">St John Stevas</surname>
</persName>
```

Values for the **type** attribute are not constrained, and may be chosen as appropriate to the encoding needs of the project. They may be used to distinguish different kinds of forename or surname, as well as to indicate the function a name component fills within the whole. In this example, we indicate that a surname is toponymic, and also point to the specific place name from which it is derived:

```
<persName>
  <forename>Johan</forename>
  <surname type="toponymic" ref="#dystvold">Dystvold</surname>
</persName>
<!-- ... -->
<placeName xml:id="dystvold">Dystvold</placeName>
```

The value **complex** was suggested above for the not uncommon case where the whole of a surname is composed of several other surname elements. These nested surnames may be individually tagged as well, together with appropriate type values:

```
<persName>
  <forename>Kara</forename>
  <surname type="complex">
    <surname type="paternal">Hattersley</surname>-
  <surname type="maternal">Smith</surname>
  </surname>
</persName>
```

The **full** attribute may be used to indicate whether a name is an abbreviation, initials, or given in full:

```
<persName>
  <forename full="abb">Maggie</forename>
  <surname>Thatcher</surname>
</persName>
```

These elements may be applied as the encoder considers appropriate, including cases where phrases or expressions are used to stand for surnames or forenames, as in the following:

```
<s>
  <persName>
    <forename>Peter</forename>
    <surname>son of Herbert</surname>
  </persName> gives the king 40 m. for
having custody of the land and heir of <persName>
    <forename>John</forename>
    <surname>son of Hugh</surname>
  </persName>...
</s>
```

Source: [28]

Similarly, patronymics may be treated as forenames, thus:

```
... but it remained for
<persName>
  <forename>Snorri</forename>
  <forename>Sturluson</forename>
</persName>
to combine the two traditions in cyclic form.
```

Source: [39]

When a patronymic is used as a surname, however (e.g. by an individual who otherwise would have no surname, but lives in a culture which requires surnames), it may be tagged as such:

```
Even <persName>
  <forename>Finnur</forename>
  <surname>Jonsson</surname>
</persName>
acknowledged the artificiality of the procedure...
```

Source: [39]

Alternatively, it may be felt more appropriate to mark a patronymic as a distinct kind of name, neither a forename nor a surname, using the <addName> element:

```
<persName>
  <forename>Egill</forename>
  <addName  type="patronym">Skallagrmsson</addName>
</persName>
```

Source: [39]

In the following example, the **type** attribute is used to distinguish a patronymic from other forenames:

```
<persName  key="pn9">
  <forename  sort="2">Sergei</forename>
  <forename  sort="3"  type="patronym">Mikhailovic</forename>
  <surname  sort="1">Uspensky</surname>
</persName>
```

This example also demonstrates the use of the **sort** attribute common to all members of the model.persNamePart class; its effect is to state the sequence in which <forename> and <surname> elements should be combined when constructing a sort key for the name.

Some names include generational or dynastic information, such as a number, or phrases such as 'Junior', or 'the Elder'; these qualifications may also be used to distinguish similarly named but unrelated people. In either case, the <genName> element may be used to distinguish such labels from other parts of the name, as in the following examples:

```
<persName  key="HEMA1">
  <surname>Marques</surname>
  <genName>Junior</genName>,
<forename>Henrique</forename>
</persName>
```

```
<persName>
  <forename>Charles</forename>
  <genName>II</genName>
</persName>
```

```
<persName>
  <forename>Rudolf</forename>
  <genName>II</genName>
  <surname  type="dynasty">Hapsburg</surname>
</persName>
```

```
<persName>
  <surname>Smith</surname>
  <genName>Minor</genName>
</persName>
```

It is also often convenient to distinguish phrases (historically similar to the generational labels mentioned above) used to link parts of a name together, such as 'von', 'of', 'de' etc. It is often a matter of arbitrary choice whether such components are regarded as part of the surname or not; the <nameLink> element is provided as a means of making clear what the correct usage should be in a given case, as in the following examples:

```
<persName  key="DUDO1">
  <roleName  type="honorific"  full="abb">Mme</roleName>
  <nameLink>de la</nameLink>
  <surname>Rochefoucault</surname>
</persName>
```

```
<persName>
  <forename>Walter</forename>
  <surname>de la Mare</surname>
</persName>
```

Finally, the <addName> and <roleName> elements are used to mark all name components other than those already listed. The distinction between them is that a <roleName> encloses an associated name component such as an aristocratic or official title which exists in some sense independently of its bearer. The distinction is not always a clear one. As elsewhere, the **type** attribute may be used with either element to supply culture- or application- specific distinctions. Some typical values for this attribute for names in the Western European tradition follow:

nobility An inherited or life-time title of nobility such as *Lord*, *Viscount*, *Baron*, etc.

honorific An academic or other honorific prefixed to a name e.g. *Doctor*, *Professor*, *Mrs.*, etc.

office Membership of some elected or appointed organization such as *President*, *Governor*, etc.

military Military rank such as *Colonel*.

epithet A traditional descriptive phrase or nick-name such as *The Hammer*, *The Great*, etc.

Note, however, that the role a person has in a given context (such as *witness*, *defendant*, etc. in a legal document) should not be encoded using the <roleName> element, since this is intended to describe the role of this part of the name, not the role of the person bearing the name. Information about roles, occupations, etc. of a person are encoded within the <person> element discussed below in *13.3. Biographical and Prosopographical Data*.

Here are some further examples of the usage of these elements:

```
<persName key="PGK1">
  <roleName type="nobility">Princess</roleName>
  <forename>Grace</forename>
</persName>
```

```
<persName key="GRMO1" type="pseudo">
  <addName type="honorific">Grandma</addName>
  <surname>Moses</surname>
</persName>
```

```
<persName key="SLWICL1">
  <roleName type="office">President</roleName>
  <forename>Bill</forename>
  <surname>Clinton</surname>
</persName>
```

```
<persName key="MOGA1">
  <roleName type="military">Colonel</roleName>
  <surname>Gaddafi</surname>
</persName>
```

```
<persName key="FRTG1">
  <forename>Frederick</forename>
  <addName type="epithet">the Great</addName>
</persName>
```

A name may have any combination of the above elements:

```
<persName key="EGBR1">
  <roleName type="office">Governor</roleName>
  <forename sort="2">Edmund</forename>
  <forename full="init" sort="3">G.</forename>
  <addName type="nick">Jerry</addName>
  <addName type="epithet">Moonbeam</addName>
  <surname sort="1">Brown</surname>
  <genName full="abb">Jr</genName>.

</persName>
```

Although highly flexible, these mechanisms for marking personal name components will not cater for every personal name and processing need. Where the internal structure of personal names is highly complex or where name components are particularly ambiguous, feature structures are recommended as the most appropriate mechanism to mark and analyze them, as further discussed in chapter *18. Feature Structures*.

13.2.2 Organizational Names

In these Guidelines, we use the term 'organization' for any named collection of people regarded as a single unit. Typical examples include businesses or institutions such as 'Harvard College' or 'the BBC', but also racial or ethnic groupings or political factions where these are regarded as forming a single agency such as 'the Scythians' or 'the Militant Tendency'. Giving a loosely-defined group of individuals a name often serves a particular political or social agenda and an analysis of the way such phrases are constructed and used may therefore be of considerable importance to the social historian, even where the objective existence of an 'organization' in this sense is harder to demonstrate than that of (say) a named person. In the case of business or other formally constituted institutions, the component parts of an organizational name may help to characterize the organization in terms of its perceived geographical location, ownership, likely number of employees, management structure, etc.

Like names of persons or places, organizational names can be marked up as referring strings or as proper names with the <rs> or <name> elements respectively. The element <orgName> is provided for use where it is desired to distinguish organizational names more explicitly.

<orgName> (organization name) contains an organizational name.

This element is a member of the same attribute classes as <persName>, as discussed above in *13.1.1. Linking Names and their Referents*.

The <orgName> element may be used to mark up any form of organizational name:

```
About a year back, a question of considerable
interest was agitated in the
<orgName type="voluntary" key="PAS1">Pennsyla. Abolition Society</orgName>
```

Source: [48]

This encoding is equivalent to, but more specific than, either of the following representations:

```
About a year back, a question of considerable
interest was agitated in the <rs key="PAS1" type="org">
  <name>Pennsyla. Abolition Society</name>
</rs>.
```

```
About a year back, a question of considerable
interest was agitated in the
<name key="PAS1" type="org">Pennsyla. Abolition
Society</name>.
```

As shown above, like the <rs> and <name> elements, the <orgName> element has a **key** attribute with which an external identifier such as a database key can be assigned to the organization name, and also a **ref** attribute which can be used to point directly to an <org> element containing information about the organization itself (see further *13.3.3. Organizational Data*). Its **type** attribute should be used to characterize the name (rather than the organization), for example as an acronym:

```
Mr Frost will be able to earn an extra fee from
<orgName type="acronym">BSkyB</orgName>
rather than the
<orgName type="acronym">BBC</orgName>
```

as a phrase:

```
The feeling in <country>Canada</country> is one of
strong aversion to the <orgName  type="phrase">United
States Government</orgName>, and of
predilection for self-government under
the
<orgName  type="phrase">English Crown</orgName>
```

Source: [198]

```
<orgName>The Justified Ancients of Mu Mu</orgName>
```

or as a composite of other kinds of name:

```
<orgName  type="partnerNames">
  <surname>Ernst</surname> & <surname>Young</surname>
</orgName>
```

The components of an organization's name are not always personal names. They may also include place names:

```
A spokesman from
<orgName  type="regional">
  <orgName>IBM</orgName>
  <country>UK</country>
</orgName> said ...
```

or role names:

```
THE TICKET which you will receive herewith has been formed by
the <orgName>Democratic Whig <name  type="role">party</name>
</orgName> after the most careful deliberation,
with a reference to all the great objects of NATIONAL, STATE,
COUNTY and CITY concern, and with a single eye to the <hi>Welfare and Best Interests of the Community</hi>.
```

As indicated above, organizational names may also be specified hierarchically particularly where the named organization is itself a department or a branch of a larger organizational entity. 'The Department of Modern History, Glasgow University' is an example:

```
<orgName>
  <orgName>Department of Modern History</orgName>
  <orgName>
    <name  type="city">Glasgow</name>
    <name  type="role">University</name>
  </orgName>
</orgName>
```

13.2.3 Place Names

Like other proper nouns or noun phrases used as names, place names can simply be marked up with the <rs> element, or with the <name> element. For cartographers and historical geographers, however, the component parts of a place name provide important information about the relation between the name and some spot in space and time. They also provide important evidence in historical linguistics.

These Guidelines distinguish three ways of referring to places. A place name (represented using the <placeName> element) may consist of one or more names for hierarchically-organized geo-political or administrative units (see section *13.2.3.1. Geo-political Place Names*). A place named simply in terms of geographical features such as mountains or rivers is represented using the <geogName> element (see section *13.2.3.2. Geographic Names*). Finally, an expression consisting of phrases expressing spatial or other kinds of relationship between other kinds of named place may itself be regarded as a way of referring to a place, and hence as a kind of named place (see section *13.2.3.3. Relative Place Names*).

<placeName> contains an absolute or relative place name.

<geogName> (geographical name) a name associated with some geographical feature such as Windrush Valley or Mount Sinai.

As members of the att.naming class, all of these elements share the following attributes:

att.naming provides attributes common to elements which refer to named persons, places, organizations etc.

- @**key** provides an external means of locating a full definition for the entity (or entities) being named, such as a database record key or other token.
- @**nymRef** (reference to the canonical name) provides a means of locating the canonical form (nym) of the names associated with the object named by the element bearing it.
- @**ref** (reference) provides an explicit means of locating a full definition for the entity being named by means of a URI.

These attributes are primarily useful as a means of linking a place name with information about a specific place. Recommendations for the encoding of information about a place, as distinct from its name, are provided in *13.3.4. Places* below.

Like the <persName> element discussed in section *13.2.1. Personal Names*, the <placeName> element may be regarded simply as an abbreviation for the elements **<name type="place">** or **<rs type="place">**. The following encodings are thus equivalent:[2]

```
After
spending some time in our <rs key="NY1" type="place">modern <name key="BA1" type="place">Babylon</name>
</rs>, <name key="NY1" type="place">New York</name>, I have proceeded to the <rs key="PH1" type="place">City
of Brotherly Love</rs>.
```

```
After spending some
time in our <placeName key="NY1">modern <placeName key="BA1">Babylon</placeName>
</placeName>, <placeName key="NY1">New
York</placeName>, I have proceeded to the <placeName key="PH1">City of
Brotherly Love</placeName>.
```

Source: [161]

[2] Strictly, a suitable value such as **figurative** should be added to the two place names which are presented periphrastically in the second example here, in order to preserve the distinction indicated by the choice of <rs> rather than <name> to encode them in the first version.

13.2.3.1 Geo-political Place Names

A place name may contain text with no indication of its internal structure:

```
<placeName>Rochester,
NY</placeName>
```

More usually however, a place name of this kind will be further analysed in terms of its constitutive geo-political or administrative units. These may be arranged in ascending sequence according to their size or administrative importance, for example: 'Rochester, New York', or as a single such unit, for example 'Belgium'. These Guidelines provide a hierarchy of generic element names, each of which may be more exactly specified by means of a **type** attribute:

<district> contains the name of any kind of subdivision of a settlement, such as a parish, ward, or other administrative or geographic unit.

<settlement> contains the name of a settlement such as a city, town, or village identified as a single geo-political or administrative unit.

<region> contains the name of an administrative unit such as a state, province, or county, larger than a settlement, but smaller than a country.

<country> contains the name of a geo-political unit, such as a nation, country, colony, or commonwealth, larger than or administratively superior to a region and smaller than a bloc.

<bloc> contains the name of a geo-political unit consisting of two or more nation states or countries.

These elements are all members of the model.placeNamePart class, members of which may be used anywhere that text is permitted, including within each other as in the following examples:

```
<placeName>
  <settlement type="city">Rochester</settlement>,
<region type="state">New York</region>
</placeName>
```

```
<placeName key="LSEA1">
  <country type="nation">Laos</country>,
<bloc type="sub-continent">Southeast Asia</bloc>
</placeName>
```

```
<placeName>
  <district type="arondissement">6ème</district>
  <settlement type="city">Paris, </settlement>
  <country>France</country>
</placeName>
```

13.2.3.2 Geographic Names

Places may also be named in terms of geographic features such as mountains, lakes, or rivers, independently of geo-political units. The <geogName> is provided to mark up such names, as an alternative to the <placeName> element discussed above. For example:

```
<geogName key="MIRI1" type="river">Mississippi River</geogName>
```

In addition to the usual phrase level elements, the <geogName> element may contain the following specialized element:

<geogFeat> (geographical feature name) contains a common noun identifying some geographical feature contained within a geographic name, such as valley, mount, etc.

Where the <geogFeat> element is used to characterize the kind of geographic feature being named, the <name> element will generally also be used to mark the associated proper noun or noun phrase:

```
<geogName key="MIRI1" type="river">
  <name>Mississippi</name>
  <geogFeat>River</geogFeat>
</geogName>
```

A more complex example, showing a variety of practices, follows:

```
The isolated ridge
separates two great corridors which run from <name key="GLCO1" type="place">Glencoe</name> into
<geogName key="GLET1" type="glen">
  <geogFeat>Glen</geogFeat>
  <name>Etive</name>
</geogName>, the
<geogName key="LAGA1" type="hill">
  <geogFeat xml:lang="gd">Lairig</geogFeat>
  <name>Gartain</name>
</geogName> and the

<geogName key="LAEI1" type="hill">
  <geogFeat xml:lang="gd">Lairig</geogFeat>
  <name>Eilde</name>
</geogName>
```

The Gaelic word *lairig* may be glossed as *sloping hill face*. The most efficient way of including this information in the above encoding would be to create a separate <nym> element for this component of the name and then point to it using the **nymRef** attribute, as further discussed in *13.3.5. Names and Nyms*.

13.2.3.3 Relative Place Names

All the place name specifications so far discussed are absolute, in the sense that they define only one place. A place may however be specified in terms of its relationship to another place, for example '10 miles northeast of Paris' or 'near the top of Mount Sinai'. These relative place names will contain a place name which acts as a referent (e.g. 'Paris' and 'Mount Sinai'). They will also contain a word or phrase indicating the position of the place being named in relation to the referent (e.g. 'the top of', 'north of'). A distance, possibly only vaguely specified, between the referent place and the place being indicated may also be present (e.g. '10 miles', 'near').

Relative place names may be encoded using the following elements in combination with either a <placeName> or a <geogName> element.

<offset> that part of a relative temporal or spatial expression which indicates the direction of the offset between the two place names, dates, or times involved in the expression.

<measure> contains a word or phrase referring to some quantity of an object or commodity, usually comprising a number, a unit, and a commodity name.

Some examples of relative place names are:

```
<placeName key="NRPA1">
  <offset>near the top of</offset>
  <geogName>
    <geogFeat>Mount</geogFeat>
    <name>Sinai</name>
  </geogName>
</placeName>
```

```
<placeName>
  <measure>20 km</measure>
  <offset>north of</offset>
  <settlement type="city">Paris</settlement>
</placeName>
```

If desired, the distance specified may be normalized using the **unit** and **quantity** attributes of <measure>:

```
<placeName key="Duncan">
  <measure unit="km" quantity="17.7">11 miles</measure>
  <offset>Northwest of</offset>
  <settlement type="city">Providence</settlement>, <region type="state">RI</region>
</placeName>
```

The internal structure of place names is like that of personal names — complex and subject to an enormous amount of variation across time and different cultures. The recommendations in this section should however be adequate for a majority of users and applications; they may be extended using the mechanisms described in chapter *23.2. Personalization and Customization* to add new elements to the existing classes. When the focus of interest is on the name components themselves, as in place name studies for example, the elements discussed in *13.3.5. Names and Nyms* may also be of use. Alternatively, the meaning structure itself may be represented using feature structures (*18. Feature Structures*).

13.3 Biographical and Prosopographical Data

This module defines a number of special purpose elements which can be used to markup biographical, historical, and prosopographical data. We envisage three basic types of users and uses for these elements. The first is the person interested in creating or converting an existing set of biographical records, for example of the type found in a Dictionary of National Biography. The second is the person hoping to create or convert a database-like collection of information about a group of people, possibly but not necessarily the people referenced in a marked-up collection of documents or a text-corpus. The third type would be those interested in the creation or conversion of biographical or CV-like structured texts for use in such applications as Human Resource management.

To cater for this diversity, these Guidelines propose a flexible approach, in which encoders may choose for themselves the degree of prescription appropriate to their needs. If one were interested, for example, in converting existing DNB-type records, and wanted to preserve the text as is, the <person> element (see *13.3.2. The Person Element*) could simply contain the text of an article, placed within <p> elements, possibly using elements such as <name> or <date> to mark up features of that text. For a more structured entry, however, one would extract the data and place information contained by the text, and encode it directly using the more specific elements described in this section.

13.3.1 Basic Principles

Information about people, places, and organizations, of whatever type, essentially comprises a series of statements or assertions relating to:

- characteristics or traits which do not, by and large, change over time
- characteristics or states which hold true only at a specific time
- events or incidents which may lead to a change of state or, less frequently, trait.

'Characteristics' or 'traits' are typically independent of an individual's volition or action and can be either physical, such as sex or hair and eye colour, or cultural, such as ethnicity, caste, or faith. The distinction is not entirely straightforward, however: while sex is fairly obviously a physical trait, gender should rather be regarded as culturally determined, and the division of mankind into different 'races', proposed by early (white European) anthropologists on the basis of physical characteristics such as skin colour, hair type and skull measurements, is by many modern cultural anthropologists now considered to be more a social or mental construct than an objective biological fact. Furthermore, while some characteristics will obviously change over time, hair colour for example, none, in principle — not even sex — is immutable.

'States' include, for example, marital status, place of residence and position or occupation. Such states have a definite duration, that is, they have a beginning and an end and are typically a consequence of the individual's own action or that of others.

By 'changes in state' are meant the events in a person's life such as birth, marriage, or appointment to office; such events will normally be associated with a specific date or a fairly narrow date-range. Changes in states can also cause or be caused by changes in characteristics. Any statement or assertion on any of these aspects of a person's life will be based on some source, possibly multiple sources, possibly contradictory. Taking all this into account it follows that each such statement or assertion needs to be able to be documented, put into a time frame and be relatable to other statements or assertions of the same or any of the other types.

The elements defined by the module described in this chapter may, for the most part, all be regarded as specialisations of one or other of the above three classes. Generic elements for state, trait, and event are also defined:

<event> contains data relating to any kind of significant event associated with a person, place, or organization.

@**where** indicates the location of an event by pointing to a <place> element

<state> contains a description of some status or quality attributed to a person, place, or organization at some specific time.

<trait> contains a description of some culturally-determined and in principle unchanging characteristic attributed to a person or place .

13.3.2 The Person Element

Information about a person, as distinct from references to a person, for example by name, is grouped together within a <person> element. Information about a group of people regarded as a single entity (for example 'the audience' of a performance) may be encoded using the <personGrp> element. Note however that information about a group of people with a distinct identity (for example a named theatrical troupe) should be recorded using the <org> element described in section *13.3.3. Organizational Data* below.

These elements may appear only within a <listPerson> element, which groups such descriptions together, and optionally also describes relationships amongst the people listed.

<person> provides information about an identifiable individual, for example a participant in a language interaction, or a person referred to in a historical source.

<personGrp> (personal group) describes a group of individuals treated as a single person for analytic purposes.

<listPerson> (list of persons) contains a list of descriptions, each of which provides information about an identifiable person or a group of people, for example the participants in a language interaction, or the people referred to in a historical source.

<relationGrp> (relation group) provides information about relationships identified amongst people, places, and organizations, either informally as prose or as formally expressed relation links.

One or more <listPerson> elements may be supplied within the <particDesc> (participant description) element in the <profileDesc> element of a TEI Header (see *2.4. The Profile Description*). Like other forms of list, however, the <listPerson> can also appear within the body of a text when the module defined by this chapter is included in a schema.

The **type** attribute may be used to distinguish lists of people of different kinds where this is considered convenient:

```
<profileDesc>
  <particDesc>
    <listPerson type="historical">
      <person xml:id="ART1">
        <persName>Arthur</persName>
      </person>
      <person xml:id="BERT1">
        <persName>Bertrand</persName>
      </person>
<!-- ... -->
    </listPerson>
    <listPerson type="mythological">
      <person xml:id="ART2">
        <persName>Arthur</persName>
      </person>
      <person xml:id="BERT2">
        <persName>Bertrand</persName>
      </person>
<!-- ... -->
    </listPerson>
  </particDesc>
</profileDesc>
```

The <person> element provides several useful attributes. First, as a member of att.editLike, the <person> element may carry attributes useful for indicating details about the scholarly interpretations made about the information recorded for the person in question:

att.editLike provides attributes describing the nature of a encoded scholarly intervention or interpretation of any kind.

- @**cert** (certainty) signifies the degree of certainty associated with the intervention or interpretation.
- @**resp** (responsible party) indicates the agency responsible for the intervention or interpretation, for example an editor or transcriber.
- @**evidence** indicates the nature of the evidence supporting the reliability or accuracy of the intervention or interpretation.
- @**source** contains a list of one or more pointers indicating the sources which support the given reading.

Second, attributes specific to <person> (and <personGrp>) allow specification of some particular information about the person (or group).

<person> provides information about an identifiable individual, for example a participant in a language interaction, or a person referred to in a historical source.

@**role** specifies a primary role or classification for the person.

@**sex** specifies the sex of the person.

@**age** specifies an age group for the person.

It is worth noting that the **age** attribute is not intended to record the person's age expressed in years, months, or other temporal unit. Rather it is intended to record into which age bracket, for the purposes of some analysis, the person falls. A simple (perhaps too simple to be useful) binary classification of age brackets would be **child** and **adult**. The actual age brackets useful to various projects are likely to be varied and idosyncratic, and thus these Guidelines make no particular recommendation as to possible values. However, it is likely to be of great value to encoders to have a closed list of possible values and documention of those values. Thus projects will typically declare the values being used in their customization file. For example, the following declaration might be useful.

```
<elementSpec ident="person" module="namesdates" mode="change">
 <attList>
  <attDef mode="replace" ident="age">
   <datatype>
    <rng:ref name="data.enumerated"/>
   </datatype>
   <valList type="closed">
    <valItem ident="child">
     <desc>less than 18 years of age</desc>
    </valItem>
    <valItem ident="adult">
     <desc>18 to 65 years of age</desc>
    </valItem>
    <valItem ident="retired">
     <desc>over 65 years of age</desc>
    </valItem>
   </valList>
  </attDef>
 </attList>
</elementSpec>
```

The above declaration, were it properly placed in a customization file, establishes that the **age** attribute of <person> has only three possible values, **child**, **adult**, and **retired**. For more information on customization see *23.2. Personalization and Customization*.

The <person> element may contain many sub-elements, each specifying a different property of the person being described. The remainder of this section describes these more specific elements. For convenience, these elements are grouped into three classes, corresponding with the tripartite division outlined above: one for traits, one for states and one for events. Each class contains both specific elements for common types of biographical information, and a generic element for other, user-defined, types of information.

All the elements in these three classes belong to the attribute class att.datable, which provides the following attributes:

att.datable.w3c provides attributes for normalization of elements that contain datable events using the W3C datatypes.

@**when** supplies the value of a date or time in a standard form, e.g. yyyy-mm-dd.

@**notBefore** specifies the earliest possible date for the event in standard form, e.g. yyyy-mm-dd.

@notAfter specifies the latest possible date for the event in standard form, e.g. yyyy-mm-dd.

@from indicates the starting point of the period in standard form, e.g. yyyy-mm-dd.

@to indicates the ending point of the period in standard form, e.g. yyyy-mm-dd.

as discussed in *13.1. Attribute Classes Defined by this Module* above.

13.3.2.1 Personal Characteristics

The model.persTraitLike class contains elements describing physical or socially-constructed characteristics or traits of a person. Members of the class comprise the following specific elements:

<faith> specifies the faith, religion, or belief set of a person.

<langKnowledge> (language knowledge) summarizes the state of a person's linguistic knowledge, either as prose or by a list of <langKnown> elements.

<langKnown> (language known) summarizes the state of a person's linguistic competence, i.e., knowledge of a single language.

<nationality> contains an informal description of a person's present or past nationality or citizenship.

<sex> specifies the sex of a person.

<age> specifies the age of a person.

<socecStatus> (socio-economic status) contains an informal description of a person's perceived social or economic status.

All, apart from <langKnowledge>, have a content model of macro.phraseSeq, by which is meant ordinary prose containing phrase-level elements.

```
<socecStatus key="AB1">Status AB1 in the RG Classification scheme</socecStatus>
```

The <langKnowledge> element contains either paragraphs or a number of <langKnown> elements; it may take a **tags** attribute, which provides one or more standard codes or 'tag's for the languages. The <langKnown> element must have a **tag** attribute, which indicates the language with the same kind of 'language tag'. These 'language tags' are discussed in detail in *vi.1 Language identification*.

Furthermore, the <langKnown> element also has a **level** attribute to indicate the level of the person's competence in the language. It is thus possible either to say:

```
<langKnowledge tags="ff fr wo en">
  <p>Speaks fluent Fulani, Wolof, and French. Some knowledge of English.</p>
</langKnowledge>
```

or

```
<langKnowledge>
  <langKnown level="fluent" tag="ff">Fulani</langKnown>
  <langKnown level="fluent" tag="wo">Wolof</langKnown>
  <langKnown level="fluent" tag="fr">French</langKnown>
  <langKnown level="basic" tag="en">English</langKnown>
</langKnowledge>
```

The <sex> element carries a **value** attribute to give the ISO 5218:1977 values (**1** for male, **2** for female, **9** for non-applicable, and **0** for unknown).

```
<sex  value="2">female</sex>
```

The generic <trait> element is also a member of this class,

<trait> contains a description of some culturally-determined and in principle unchanging characteristic attributed to a person or place .

This element can be used to extend the range of information supplied about an individual's personal traits. It may contain an optional <label> element, used to provide a human-readable specification for the category of trait or feature concerned and a description of the feature itself supplied within a <desc> element. These may be followed by or one or more <p> elements supplying more detailed information about the trait. In either case, these may be followed by one or more notes or bibliographical references. The **type**, **ref**, and **key** attributes are available on the <trait> element to indicate a fuller definition of the combination of feature and value.

```
<trait  type="ethnicity"  key="alb">
  <label>Ethnicity</label>
  <desc>Ethnic Albanian.</desc>
</trait>
```

The generic element can be used in place of one of the more specific elements:

```
<trait  type="nationality"  notBefore="2002-01-15">
  <label>Nationality</label>
  <desc>American citizen from 15 January 2002.</desc>
</trait>
```

is the same as:

```
<nationality  notBefore="2002-01-15">Became an American citizen on 15 January 2002.</nationality>
```

or even:

```
<nationality  notBefore="2002-01-15"  key="US"/>
```

More usually however, the element is provided as a simple means of extending the set of descriptive features available in a standardized way. For example, there are no predefined elements for such features as eye or hair colour. If these are to be recorded, they may simply be added as new types of trait:

```
<trait  type="physical">
  <label>eye colour</label>
  <desc>blue</desc>
</trait>
<trait  type="physical">
  <label>hair colour</label>
  <desc>brown</desc>
</trait>
```

13.3.2.2 Personal States

The model.persStateLike class contains elements describing changeable characteristics of a person which have a definite duration, for example occupation, residence, or name. Members of the class comprise the following specific elements:

<persName> (personal name) contains a proper noun or proper-noun phrase referring to a person, possibly including any or all of the person's forenames, surnames, honorifics, added names, etc.

<occupation> contains an informal description of a person's trade, profession or occupation.

<residence> (residence) describes a person's present or past places of residence.

<affiliation> contains an informal description of a person's present or past affiliation with some organization, for example an employer or sponsor.

<education> contains a description of the educational experience of a person.

<floruit> contains information about a person's period of activity.

The <persName> element is repeatable and can, like all TEI elements, take the attribute **xml:lang** to indicate the language of the content of the element, as well as a **type** attribute to indicate the type of name, whether a nickname, maiden name, alternative form, etc. This is useful in cases where, for example, a person is known by a Latin name and also by any number of vernacular names, many or all of which may have claims to 'authenticity'. In order to ensure uniformity, the method generally employed in the library world has been to accept the form found in some authority file, for example that of the American Library of Congress, as the 'base' or 'neutral' form. Feelings can run high on this matter, however, and people are often reluctant to accept as 'neutral' an overtly foreign form of the name of their local saint or hero. Within the <person> element any number of variant forms of a name can be given, with no prioritisation, and hence less likelihood of offence. The Icelandic scholar and manuscript collector Árni Magnússon, to give his name in standard modern Icelandic spelling, is known in Danish as Arne Magnusson, the form which he himself, as a life-long resident of Denmark, generally used; there is also a Latinised form, Arnas Magnæus, which he used in his scholarly writings. All three forms can be given, and in any order:

```
<person xml:id="ArnMag">
  <persName xml:lang="is">Árni Magnússon</persName>
  <persName xml:lang="da">Arne Magnusson</persName>
  <persName xml:lang="la">Arnas Magnæus</persName>
</person>
```

At the other extreme, a person may be named periphrastically as in the following example:

```
<person xml:id="simon_son_of_richard2">
  <persName>Simon, son of Richard</persName>
  <residence>
    <placeName>
      <region>Essex</region>
    </placeName>
  </residence>
  <floruit notBefore="1219" notAfter="1223">1219-1223</floruit>
</person>
```

In addition to these specific elements the class contains a generic element called <state>.

<state> contains a description of some status or quality attributed to a person, place, or organization at some specific time.

This element can be used to extend the range of descriptive information available in the same way as the <trait> element, using the same content model. For example, a description of the first living held by the Icelandic clergyman and poet Jón Oddsson Hjaltalín might be tagged as follows:

```
<state type="office" from="1777-04-07" to="1780-07-12">
  <p>Jón's first living – which he apparently accepted rather reluctantly – was at
  <name type="place">Háls í Hamarsfirði</name>, <name type="place">Múlasýsla</name>, to which
     he was presented on 7 April 1777. He was ordained the following
     month and spent three years at Háls, but was never happy there,
     due largely to the general penury in which he was forced to live –
     a recurrent theme throughout the early part of his life. In June
     of 1780 the bishop recommended that Jón
     should <q xml:lang="da">promoveres til andet bedre kald, end det
        hand hidindtil har havt</q>, and on 12 July it was agreed that
     he should exchange livings with
  <name type="person" key="ThorJon">sr. Þórður Jónsson</name> at
  <name type="place">Kálfafell á Síðu</name>,
  <name type="place">Skaftafellssýsla</name>.</p>
  <bibl>ÞÍ, Stms I.15, p. 733.</bibl>
  <bibl>ÞÍ, Stms I.17, p. 102.</bibl>
</state>
```

13.3.2.3 Personal Events

The model.persEventLike class contains elements describing specific events in a person's history, for example birth, marriage, or appointment. These are not characteristics of an individual, but often cause an individual to gain such characteristics, or to enter a new state. Members of this class comprise the following elements:

<birth> contains information about a person's birth, such as its date and place.

<death> contains information about a person's death, such as its date and place.

Only two specific elements (<birth> and <death>) are proposed. The generic element <event> has a similar content model to that of <state> and <trait>; the chief difference being that it can include a <placeName> element to identify the name of the place where the event occurred. It is used to describe any event in the life of an individual or organization.

In the following example, we give a brief summary of the wedding of Jane Burden to the English writer, designer, and socialist William Morris, encoded as an <event> element embedded within the <person> element used to record data about Morris, though we could equally well have embedded the event within the <person> element for Burden, or have given it as a freestanding <event> independent of either <person> element:

```
<person xml:id="WM">
<!-- ... -->
  <event type="marriage" when="1859-04-26">
    <label>Marriage</label>
    <desc>
      <name type="person" ref="#WM">William Morris</name> and <name type="person" ref="#JBM">Jane Bur-
den</name> were
        married at <name type="place">St Michael's Church, Ship Street, Oxford</name> on
    <date when="1859-04-26">26 April 1859</date>. The wedding was
        conducted by Morris's friend <name type="person" ref="#RWD">R. W.
           Dixon</name> with <name type="person" ref="#CBF">Charles
           Faulkner</name> as
        the best man. The bride was given away by her father,
    <name type="person" ref="#RB">Robert Burden</name>.
        According to the account that <name type="person" ref="#EBJ">Burne-Jones</name>
```

```
        gave <name  type="person"  ref="#JWM">Mackail</name>
     <quote>M. said to Dixon beforehand <said>Mind
              you don't call her Mary</said> but he did</quote>. The entry in the
        Register reads: <quote>William Morris, 25, Bachelor Gentleman, 13
           George Street, son of William Morris decd. Gentleman. Jane Burden,
           minor, spinster, 65 Holywell Street, d. of Robert Burden,
           Groom.</quote> The witnesses were Jane's parents and Faulkner. None of
        Morris's family attended the ceremony. Morris presented Jane with a
        plain gold ring bearing the London hallmark for 1858. She gave her
        husband a double-handled antique silver cup.</desc>
    <bibl>J. W. Mackail, <title>The Life of William Morris</title>, 1899.</bibl>
  </event>
</person>
<person  xml:id="JBM">
  <persName>Jane Burden</persName>
</person>
<person  xml:id="RWD">
  <persName>R.W. Dixon</persName>
</person>
<person  xml:id="CBF">
  <persName>Charles Faulkner</persName>
</person>
<person  xml:id="EBJ">
  <persName>
    <forename>Edward</forename>
    <surname>Burne-Jones</surname>
  </persName>
</person>
<person  xml:id="JWM">
  <persName>J.W. Mackail</persName>
</person>
```

In this example the **ref** attributes on the various <name> elements point to the <person> elements for the other people named. As further discussed below (*13.3.2.4. Personal Relationships*), a <relation> element may then be used to link them in a more meaningful way:

```
<relation  name="spouse"  mutual="#WM #JBM"/>
<relation  name="friend"  mutual="#WM #RWD"/>
<relation  name="parent"  active="#RB"  passive="#JBM"/>
```

As mentioned above, all these elements, both the specific and the generic, are members of the att.datable attribute class, which means they can be limited in terms of time. The following encoding, for example, demonstrates that the person named David Jones changed his name in 1966 to David Bowie:

```
<person  xml:id="DB">
  <persName  notAfter="1966">David Jones</persName>
  <persName  notBefore="1966">David Bowie</persName>
</person>
```

All the generic elements are also members of the att.editLike class, which, as its name implies, was originally intended to provide attributes 'describing the nature of an encoded scholarly intervention or interpretation of any kind' and which makes available the attributes **cert**, to indicate the degree of certainty, **resp**, the agency

responsible, and **evidence**, the nature of the evidence used. In this way it is possible, in the case of multiple and conflicting sources, to provide more than one view of what happened, as in the following example:

```
<event type="birth" resp="#XYZ" cert="high">
  <p>Born in <name type="place">Brixton</name> on 8 January 1947.</p>
</event>
<event type="birth" resp="#ABC" cert="low">
  <p>Born in <name type="place">Berkhamsted</name> on 9 January 1947.</p>
</event>
```

13.3.2.4 Personal Relationships

When the module defined by this chapter is included in a schema, the following two elements may be used to document relationships amongst the persons, places, or organizations identified:

<relationGrp> (relation group) provides information about relationships identified amongst people, places, and organizations, either informally as prose or as formally expressed relation links.

<relation> (relationship) describes any kind of relationship or linkage amongst a specified group of participants.

@**name** supplies a name for the kind of relationship of which this is an instance.

@**active** identifies the 'active' participants in a non-mutual relationship, or all the participants in a mutual one.

@**mutual** supplies a list of participants amongst all of whom the relationship holds equally.

@**passive** identifies the 'passive' participants in a non-mutual relationship.

These elements are both members of the att.typed class, from which they inherit the **type** and **subtype** attributes in the usual way. The value specified for either attribute on a <relationGrp> element is implicitly applicable to all of its child <relation> elements, unless overriden.

A relationship, as defined here, may be any kind of describable link between specified participants. A participant (in this sense) might be a person, a place, or an organization. In the case of persons, therefore, a relationship might be a social relationship (such as employer/employee), a personal relationship (such as sibling, spouse, etc.) or something less precise such as 'possessing shared knowledge'. A relationship may be mutual, in that all the participants engage in it on an equal footing (for example the 'sibling' relationship); or it may not be if participants are not identical with respect to their role in the relationship (for example, the 'employer' relationship). For non-mutual relationships, only two kinds of role are currently supported; they are named active and passive. These names are chosen to reflect the fact that non-mutual relations are directed, in the sense that they are most readily described by a transitive verb, or a verb phrase of the form *is X of* or *is X to*. The subject of the verb is classed as active; the direct object of the verb, or the object of the concluding preposition, as passive. Thus parents are 'active' and children 'passive' in the relationship 'parent' (interpreted as *is parent of*); the employer is 'active', the employee 'passive', in the relationship *employs*. These relationships can be inverted: parents are 'passive' and children 'active' in the relationship *is child of*; similarly 'works for' inverts the active and passive roles of 'employs'.

For example:

```
<relationGrp>
  <relation name="parent" active="#P1 #P2" passive="#P3 #P4"/>
  <relation name="spouse" mutual="#P1 #P2"/>
  <relation
     type="social"
     name="employer"
```

```
    active="#P1"
    passive="#P3 #P4"/>
</relationGrp>
```

This example defines the relationships amongst a number of people not further described here; we assume however that each person has been allocated an identifier such as **P1**, **P2**, etc. which can be linked to using the reference **#P1**. Then the above set of <relation> elements describe the following three relationships amongst the seven people referenced:

- P1 and P2 are parents of P3 and P4.
- P1 and P2 are linked in a mutual relationship called 'spouse' — that is, P2 is the spouse of P1, and P1 is the spouse of P2.
- P1 has the social relationship 'employer' with respect to P3, and P4.

Relationships within places and organizations are further discussed in the relevant section below. Relationships between for example organizations and places, or places and persons, may be handled in exactly the same way.

13.3.3 Organizational Data

The <org> and <listOrg> elements are used to store data about an organization such as its preferred name, its locations, or key persons within it.

<org> (organization) provides information about an identifiable organization such as a business, a tribe, or any other grouping of people.

<listOrg> (list of organizations) contains a list of elements, each of which provides information about an identifiable organization.

These elements are intended to be used in a way analogous to the <place> and <person> elements discussed elsewhere in this chapter, that is to provide as a unique wrapper element for information about an entity, distinct from references to that entity which are typically encoded using a naming element such as **<name type="org">** or <orgName>. The content of a naming element will represent the way an organization is named in a given context; the content of an <org> represents the information known to the encoder about that organization, gathered together in a single place, and independent of its textual realization.

An organization is not the same thing as a list or group of people because it has an identity of its own. That identity may be expressed solely in the existence of a name (for example 'The Scythians'), but is likely to consist in the combination of that name with a number of events, traits, or states which are considered to apply to the organization itself, rather than any of its members. For example, a sports team might be defined in terms of its membership (a <listPerson>), its fixtures (a <listPlace>), its geographical affiliation (a <placeName>), or any combination of these. It will also have properties which may be used to categorize it in some way such as the kind of sport played, whether the team is amateur or professional, and so on: these are probably best dealt with by means of the **type** attribute. However, it is the name of the sports team alone which identifies it.

The content model for <org> permits any mixture of generic <state>, <trait>, or <event> elements: the presence of the <orgName> element described in *13.2.2. Organizational Names* is however strongly recommended.

In other respects, the <org> element is used in much the same way as <place> or <person>. An organization may have different names at different times:

```
<org>
  <orgName  notAfter="1960">The Silver Beetles</orgName>
```

```
  <orgName  from="1960-08">The Beatles</orgName>
</org>
```

The names of the people making up an organization can also change over time, (if they are known at all). For example:

```
<org>
  <orgName  notAfter="1960">The Silver Beetles</orgName>
  <orgName  notBefore="1960">The Beatles</orgName>
  <state  type="membership"  from="1960-08"  to="1962-05">
    <desc>
      <persName>John Lennon</persName>
      <persName>Paul McCartney</persName>
      <persName>George Harrison</persName>
      <persName>Stuart Sutcliffe</persName>
      <persName>Pete Best</persName>
    </desc>
  </state>
  <state  type="membership"  notBefore="1963">
    <desc>
      <persName>John Lennon</persName>
      <persName>Paul McCartney</persName>
      <persName>George Harrison</persName>
      <persName>Ringo Starr</persName>
    </desc>
  </state>
</org>
```

An <org> may contain subordinate <org>s:

```
<org>
  <orgName>Oxford University Computing Services</orgName>
  <org>
    <orgName>Information and Support Group</orgName>
  </org>
  <org>
    <orgName>Infrastructure Group</orgName>
    <org>
      <orgName>Networking Team</orgName>
    </org>
    <org>
      <orgName>System Development Team</orgName>
    </org>
  </org>
  <org>
    <orgName>Learning Technologies Group</orgName>
  </org>
</org>
```

The following example demonstrates the use of the <listOrg> element to group together a number of <org> elements, each of which is defined solely by means of an informal description, itself containing other names.

```
<p>The TEI institutional hosts are: <listOrg>
   <org  xml:id="bu">
     <orgName>Brown University</orgName>
     <desc>The host contribution is made jointly by the <name  type="project">Brown University Women Writers
Project</name> and the
     <orgName>Brown University Library's Center for Digital
              Initiatives</orgName>.</desc>
   </org>
   <org  xml:id="na">
     <orgName>Nancy</orgName>
     <desc>Hosting is provided by a group of institutions located in
           Nancy, France, coordinated by <orgName>Loria</orgName> and also
           including <orgName>ATILF</orgName> and <orgName>INIST</orgName>.</desc>
   </org>
   <org  xml:id="ou">
     <orgName>Oxford University</orgName>
     <desc>Hosting is provided by the <orgName>Research Technologies
              Service</orgName> at <orgName>Oxford University Computing
              Services</orgName>.</desc>
   </org>
   <org  xml:id="uv">
     <orgName>University of Virginia</orgName>
     <desc>Virginia's host support comes jointly from the
     <orgName>Institute for Advanced Technology in the
              Humanities</orgName> and the <orgName>University of Virginia
              Library</orgName>.</desc>
   </org>
<!-- from http://www.tei-c.org/About/hosting.xml-->
  </listOrg>
</p>
```

In a more elaborated version of this example, the organizational names tagged using <orgName> might be linked using the **key** or **ref** attribute to a unique <org> element elsewhere.

13.3.4 Places

In *13.2.3. Place Names* we discuss various ways of naming places such as towns, countries, etc. In much the same way as these Guidelines distinguish between the encoding of names for people and the encoding of other data about people, so they also distinguish between the encoding of names for places and the encoding of other data about places. In this section we present elements which may be used to record in a structured way data about places of any kind which might be named or referenced within a text. Such data may be useful as a way of normalising or standardizing references to particular places, as the raw material for a gazetteer or similar reference document associated with a particular text or set of texts, or in conjunction with any form of geographical information system.

The following elements are provided for this purpose:

<listPlace> (list of places) contains a list of places, optionally followed by a list of relationships (other than containment) defined amongst them.

<place> contains data about a geographic location

The model.placeStateLike class contains elements describing characteristics of a place which have a definite duration, such as its name. Any member of the model.placeNamePart may be used for this purpose, since a <place> element will usually contain at least one, and possibly several, <placeName>-like elements indicating the names associated with it, by different people, in different languages, or at different times.

For example, the modern city of Lyon in France was in Roman times known as Lugdunum. Although the modern and the Roman city are not physically co-extensive, they have significant areas which overlap, and we may therefore wish to regard them as the same place, while supplying both names with an indication of the time period during which each was current.

A place is defined, however, by its physical location, which does not typically change over time; we regard the location therefore as a trait of the place, and represent it by means of elements from the model.placeTraitLike class. Locations may be specified in a number of ways: as a set of coordinates defining a point or an area on the surface of the earth, or by providing a description of how the place may be found, usually in terms of other place names. For example, we can identify the location of the Canadian city of London, either by specifying its latitute and longitude, or by specifying that we mean the city called London located in the province called Ontario within the country called Canada.

In addition we may wish to supply a brief characterization of the place identified, for example to state that it is a city, an administrative area such as a country, or a landmark of some kind such as a monument or a battlefield. If our typology of places is simple, the open ended **type** attribute is the easiest way to represent it: so we might say **<place type="city">**, **<place type="battlefield">** etc.

Within the <place> element, the following elements may be used to provide more information about specific aspects of the place in a structured form:

<placeName> contains an absolute or relative place name.

<location> defines the location of a place as a set of geographical coordinates, in terms of a other named geo-political entities, or as an address.

13.3.4.1 Varieties of Location

A location may be specified in one or more of the following ways:

1. by supplying a string representing its coordinates in some standardized way within a <geo> element, as shown below
2. by supplying one or more place name component elements (e.g. <country>, <settlement> etc.) to place it within a geo-political context
3. by supplying a postal address, e.g. using the <address> element
4. by supplying a brief textual description, e.g. using the <desc> element
5. by using a non-TEI XML vocabulary such as the Geography Markup Language

We give examples of all of these methods in the remainder of this section.

The simplest method of specifying a location is by means of its geographic coordinates, supplied within the <geo> element. This may be used to supply any kind of positional information, using one of the many different geodetic systems available. Such systems vary in their format, in their scope or coverage, and more fundamentally in the reference frame (the 'datum') used for the coordinate system itself. The default recommended by these Guidelines is to supply a string containing two real numbers separated by whitespace, of which the first indicates latitude and the second longitude according to the 1984 World Geodetic System (WGS84); this is the system currently used by most GPS applications which TEI users are likely to encounter.[3]We might therefore record the information about the place known as 'Lyon' as follows:

[3]See `http://earth-info.nga.mil/GandG/wgs84/index.html`. The most recent revision of this standard is known as the Earth Gravity Model 1996.

```
<place xml:id="LYON1" type="city">
  <placeName notBefore="1400">Lyon</placeName>
  <placeName notAfter="0056">Lugdunum</placeName>
  <location>
    <geo>41.687142 -74.870109</geo>
  </location>
</place>
```

Identifying Lyon by its geo-political status as a settlement within a country forming part of a larger political entity, we might represent the same 'place' as follows:

```
<place xml:id="LYON2">
  <placeName notBefore="1400">Lyon</placeName>
  <placeName notAfter="0056">Lugdunum</placeName>
  <location>
    <bloc>EU</bloc>
    <country>France</country>
  </location>
</place>
```

Elements such as <bloc> are specialised forms of <placeName>, as discussed in *13.2.3.1. Geo-political Place Names*.

We may use the same procedure to represent the location of smaller places, such as a street or even an individual building:

```
<place type="building">
  <placeName>Brasserie Georges</placeName>
  <location>
    <country key="FR"/>
    <settlement type="city">Lyon</settlement>
    <district type="arrondissement">Perrache</district>
    <placeName type="street">Rue de la Charité</placeName>
  </location>
</place>
```

Note the use of the **type** attribute to categorize more precisely both the kind of place concerned (a building) and the kind of name used to locate it, for example by characterizing the generic <district> as an 'arrondissement'.

We may even wish to treat imaginary places in the same way:

```
<place type="imaginary">
  <placeName>Atlantis</placeName>
  <location>
    <offset>beyond</offset>
    <placeName>The Pillars of <persName>Hercules</persName>
    </placeName>
  </location>
</place>
```

A <location> sometimes resembles a set of instructions for finding a place, rather than a name:

```
<place xml:id="MYF">
  <placeName notAfter="1969">Yasgur's Farm</placeName>
  <placeName notBefore="1969">Woodstock Festival Site</placeName>
  <location>
    <measure>one mile</measure>
    <offset>north west of</offset>
    <settlement>Bethel</settlement>
    <region>New York</region>
  </location>
</place>
```

The element <address> may also be used to identify a location in terms of its postal or other address:

```
<place type="cemetery">
  <placeName>Protestant Cemetery</placeName>
  <placeName type="official" xml:lang="it">Cimitero Acattolico</placeName>
  <location type="geopolitical">
    <country>Italy</country>
    <settlement>Rome</settlement>
    <district>Testaccio</district>
  </location>
  <location type="address">
    <address>
      <addrLine>Via Caio Cestio, 6</addrLine>
      <addrLine>00153 Roma</addrLine>
    </address>
  </location>
</place>
```

When, as here, the same place is given multiple locations, the **type** attribute should be used to characterize the kind of location, as a means of indicating that these are alternative ways of identifying the same place, rather than that place is spread across several locations.

The <location> element may thus identify a place to a greater or lesser degree of precision, using a variety of means: a name, a set of names, or a set of coordinates. The <geo> element introduced earlier is by default understood to supply a value expressed in a specific (and widely used) notation; this may be modified in two ways.

Firstly, the content of the <geo> element could be interpreted in some other way, that is, according to some different geodetic system. By default, a standard known as the World Geodetic System (WGS) is employed; the following element is provided to indicate (within the header of a document) a different notation, or one based on a different datum, has been employed:

<geoDecl> (geographic coordinates declaration) documents the notation and the datum used for geographic coordinates expressed as content of the <geo> element elsewhere within the document.

@**datum** supplies a commonly used code name for the datum employed.

Secondly, the element <geo> may be redefined to contain markup from a different XML vocabulary which is specifically designed to represent this kind of information. This technique is used throughout the Guidelines where specialized markup is required, for example to embed mathematical expressions or vector graphics, and is further described and exemplified in *23.2.4. Examples of Modification* . For geographic information, suitable non-TEI vocabularies include:

- the Geographical Markup Language (GML) being defined by the OGC[4]
- the Keyhole Markup Language (KML) now used by Google Maps[5]

In the following example, we have defined the location of the place 'Lyon' using GML and indicated the two names associated with it at different times:

```
<place type="city">
<placeName notBefore="1400">Lyon</placeName>
<placeName notAfter="0056">Lugdunum</placeName>
<location>
<geo>
<gml:Polygon>
<gml:exterior>
<gml:LinearRing> 45.256 -110.45 46.46 -109.48 43.84 -109.86 45.8 -109.2
45.256 -110.45 </gml:LinearRing>
</gml:exterior>
</gml:Polygon></geo>
</location>
</place>
```

A <bibl> element may be used within <location> to indicate the source of the location information.

13.3.4.2 Multiple Places

A place may contain other places. This containment relation can be directly modelled in XML: thus we can say that the towns of Vilnius and Kaunas are both in a place called Lithuania (or Lietuva) as follows:

```
<place>
  <country>Lithuania</country>
  <country  xml:lang="lt">Lietuva</country>
  <place>
    <settlement>Vilnius</settlement>
  </place>
  <place>
    <settlement>Kaunas</settlement>
  </place>
</place>
```

This does not, of course, imply that Vilnius and Kaunas are the only places constituting Lithuania; only that they are within it. A separate <place> element may indicate that it is a part of Lithuania by supplying a <relation> element, as discussed below (*13.3.4.4. Relations Between Places*).

As a further example, the islands of Mauritius, Réunion, and Rodrigues are collectively known as the Mascarene Islands. Grouped together with Mauritius there are also several smaller offshore islands, with rather picturesque French names. These offshore islands do not however constitute an identifiable place as a whole. One way of representing this is as follows:

```
<place  type="islandGroup">
  <placeName>Mascarene Islands</placeName>
  <placeName>Mascarenhas Archipelago</placeName>
```

[4] The OGC is an international voluntary consensus standards organization whose members maintain the Geography Markup Language standard. The OGC coordinates with the ISO TC 211 standards organization to maintain consistency between OGC and ISO standards work. GML is in the process of being adopted as an ISO standard (ISO 19136) and is expected to be released as an International Standard in 2007.

[5] See http://code.google.com/apis/kml/documentation/index.html

```
  <place type="island">
    <placeName>Mauritius</placeName>
    <listPlace type="offshoreIslands">
      <place>
        <placeName>La roche qui pleure</placeName>
      </place>
      <place>
        <placeName>Ile aux cerfs</placeName>
      </place>
    </listPlace>
  </place>
  <place type="island">
    <placeName>Rodrigues</placeName>
  </place>
  <place type="island">
    <placeName>Réunion</placeName>
  </place>
</place>
```

Here is a more complex example, showing the variety of names associated at different times and in different languages with a set of hierarchically grouped places — the settlement of Carmarthen Castle, within the town of Carmarthen, within the administrative county of Carmarthenshire, Wales.

```
<place xml:id="wales" type="country">
  <placeName xml:lang="cy">Cymru</placeName>
  <placeName xml:lang="en">Wales</placeName>
  <placeName xml:lang="la">Wallie</placeName>
  <placeName xml:lang="la">Wallia</placeName>
  <placeName xml:lang="fro">Le Waleis</placeName>
  <place xml:id="carmarthenshire" type="region">
    <region type="county" xml:lang="en" notBefore="1284">Carmarthenshire</region>
    <place xml:id="carmarthen" type="settlement">
      <placeName xml:lang="en">Carmarthen</placeName>
      <placeName xml:lang="la" notBefore="1090" notAfter="1300">Kaermerdin</placeName>
      <placeName xml:lang="cy">Caerfyrddin</placeName>
      <place xml:id="carmarthen_castle" type="castle">
        <settlement>castle of Carmarthen</settlement>
      </place>
    </place>
  </place>
</place>
```

As noted previously, <country>, <region>, and <settlement> are all specializations of the generic <placeName> element; they are not specializations of the <place> element. If it is desired to distinguish amongst kinds of *place* this can only be done by means of the **type** attribute as in the above example.

This use of multiple <place> elements should be distinguished from the (possibly simpler) case where a number of places with some property in common are being grouped together for convenience, for example, in a gazetteer. The <listPlace> element is provided as a means of grouping places together where there is no implication that the grouped elements constitute a distinct place. For example:

```
<place type="county">
  <placeName>Herefordshire</placeName>
```

```
  <listPlace type="villages">
    <place>
      <placeName>Abbey Dore</placeName>
      <location>
        <geo>51.969604 -2.893146</geo>
      </location>
    </place>
    <place>
      <placeName>Acton Beauchamp</placeName>
    </place>
<!-- etc -->
  </listPlace>
  <listPlace type="towns">
    <place>
      <placeName>Hereford</placeName>
    </place>
    <place>
      <placeName>Leominster</placeName>
    </place>
<!-- etc -->
  </listPlace>
</place>
```

13.3.4.3 States, Traits, and Events

There are many different kinds of information which it might be considered useful to record for a place in addition to its name and location, and the categories selected are likely to be very project-specific. As with persons therefore these Guidelines make no claim to comprehensiveness in this context. Instead, the generic <state>, <trait>, and <event> elements defined by this module should be used. Each of these may be customized for particular needs by means of their **type** attribute. These are complemented by a small number of predefined elements of general utility:

<population> contains information about the population of a place.

<climate> contains information about the physical climate of a place.

<terrain> contains information about the physical terrain of a place.

These are all specializations of the generic <trait> element. This element may be used for almost any kind of event in the life of a place; no specialized version of this element is proposed, nor do we attempt to enumerate the possible values which might be appropriate for the **type** attribute on any of these generic elements.

Here is an example, showing how the specific and generic elements may be combined:

```
<place xml:id="IS">
  <placeName xml:lang="en">Iceland</placeName>
  <placeName xml:lang="is">Ísland</placeName>
  <location>
    <geo>65.00 -18.00</geo>
  </location>
  <terrain>
    <desc>Area: 103,000 sq km</desc>
  </terrain>
  <state type="governance" notBefore="1944">
    <p>Constitutional republic</p>
  </state>
  <state type="governance" notAfter="1944">
    <p>Part of the kingdom of <placeName key="DK">Denmark</placeName>
```

```
    </p>
  </state>
  <event  type="governance"  when="1944-06-17">
    <desc>Iceland became independent on 17 June 1944.</desc>
  </event>
  <state  type="governance"  from="1944-06-17">
    <p>An independent republic since June 1944</p>
  </state>
</place>
```

In the following example, the <climate> example is used to provided a detailed discussion of this particular aspect of the information available about a particular place:

```
<place  xml:id="greece">
  <placeName>Greece</placeName>
  <climate>
    <desc>Greece's climate is divided into three well defined
        classes:</desc>
    <climate>
      <label>Mediterranean</label>
      <desc>It features mild, wet winters and hot, dry
           summers. Temperatures rarely reach extremes, although snowfalls do
           occur occasionally even in <placeName>Athens</placeName>,
      <placeName>Cyclades</placeName> or <placeName>Crete</placeName>
           during the winter.</desc>
    </climate>
    <climate>
      <label>Alpine</label>
      <desc>It is found primarily in <placeName>
         <offset>Western</offset>
              Greece</placeName> (<placeName>Epirus</placeName>,
      <placeName>
         <offset>Central</offset> Greece</placeName>,
      <placeName>Thessaly</placeName>,
      <placeName>
         <offset>Western</offset> Macedonia</placeName> as well
           as central parts of <placeName>Peloponnesus</placeName> like
      <placeName>Achaea</placeName>, <placeName>Arcadia</placeName> and
           parts of <placeName>Laconia</placeName> where the Alpine range pass
           by)</desc>
    </climate>
    <climate>
      <label>Temperate</label>
      <desc>It is found in <placeName>
         <offset>Central</offset> and
        <offset>Eastern</offset> Macedonia</placeName> as well as in
      <placeName>Thrace</placeName> at places like
      <placeName>Komotini</placeName>, <placeName>Xanthi</placeName> and
      <placeName>
         <offset>northern</offset> Evros</placeName>. It features
           cold, damp winters and hot, dry summers.</desc>
    </climate>
  </climate>
</place>
```

As the above exanmple shows, <state> and <trait> elements, and others of the same class, can be nested hierarchically within each other. When this is done, values for the **type** attribute are to be understood as cumulatively inherited, as elsewhere in the TEI scheme (for example on <category> or <linkGrp>). In the following example, the outermost <population> element concerns the squirrel population between the dates given. This is then broken down into red and gray squirrel populations, and within that into male and female:

```
<population
   type="squirrel"
   notBefore="1901"
   notAfter="1902-01-11"
   resp="#strabo">
  <population type="red" when="1901-01-10">
   <population type="female">
    <desc>12</desc>
   </population>
   <population type="male">
    <desc>15</desc>
   </population>
  </population>
  <population type="gray" when="1902-01-10" cert="high">
   <population type="female">
    <desc>23</desc>
   </population>
   <population type="male" cert="low" resp="#biber">
    <desc>45</desc>
   </population>
  </population>
</population>
```

The dating and responsibility attributes here behave slightly differently from the **type** attribute: responsibility is not an additive property, and therefore an element either states it explicitly, or inherits it from its nearest ancestor. Dating is slightly different again, in that a child element may specify a date more precisely than its parent, as in the example above

Events may also be subdivided into other events. For example, a two part meeting might be represented as follows:

```
<event type="meeting" when="2007-05-29">
  <desc>All day meeting to resolve content models</desc>
  <event type="preamble" notAfter="13:00:00">
   <desc>first part</desc>
  </event>
  <event type="conclusions" notBefore="13:00:00">
   <desc>second part</desc>
  </event>
</event>
```

13.3.4.4 Relations Between Places

The <relation> element may also be used to express relationships of various kinds between places, or between places and persons, in much the same way as it is used to express relationships between persons alone. Returning to the Mascarene Islands example cited above, we might define the island group and its constituents separately, but indicate the relationship by means of a <relation> element:

```
<listPlace>
  <place xml:id="MASC">
    <placeName>Mascarene islands</placeName>
    <placeName>Mascarenhas Archipelago</placeName>
  </place>
  <place xml:id="MRU">
    <placeName>Mauritius</placeName>
<!-- ... -->
  </place>
  <place xml:id="ROD">
    <placeName>Rodrigues</placeName>
  </place>
  <place xml:id="REN">
    <placeName>Réunion</placeName>
  </place>
  <relation name="contains" active="#MASC" passive="#ROD #MRU #REN"/>
</listPlace>
```

This 'stand off' style of representation has the advantage that we can now also represent the fact that a place may be a 'part of' more than one other place; for example, Réunion is part of France, as well as part of the Mascarenes. If we add a declaration for France to the list above:

```
<place type="country" xml:id="FRA">
  <placeName>France</placeName>
</place>
```

we can now model this dual allegiance by means of a <relation> element:

```
<relation name="partOf" active="#REN" passive="#FRA #MASC"/>
```

13.3.5 Names and Nyms

So far we have discussed ways in which a name or referring string encountered in running text may be resolved by considering the object that the name refers to: in the case of a personal name, the name refers to a person; in the case of a place name, to a place, for example. The resolution of this reference is effected by means of the **key** or **ref** attributes available to all elements which are members of the att.naming class, such as <persName> or <placeName> and their more specialized variants such as <forename> or <country>. However, *names* can also be regarded as objects in their own right, irrespective of the objects to which they are attached, notably in onomastic studies. From this point of view, the names *John* in English, Jean in French, and Ivan in Russian might all be regarded as existing independently of any person to which they are attached, and also independently of any variant forms that might be attested in different sources (such as Jon or Johnny in English, or Jehan or Jojo in French). We use the term nym to refer to the canonical or normalized form of a name regarded in such a way, and provide the following elements to encode it:

<listNym> (list of canonical names) contains a list of nyms, that is, standardized names for any thing.

<nym> (canonical name) contains the definition for a canonical name or namepart of any kind.

Any element which is a member of the att.naming class may use the attribute **nymRef** to indicate the nym with which it corresponds. Thus, given the following <nym> for the name *Antony*:

```
<listNym>
  <nym xml:id="N123">
    <form>Antony</form>
  </nym>
<!-- other nym definitions here -->
</listNym>
```

an occurrence of this name in running text might be encoded as follows:

```
<forename nymRef="#N123">Tony</forename> Blair
```

Note that this association (between "Tony" and "Antony") has nothing to do with any individual who might use the name.

The person identified by this particular Tony may however be indicated independently using the **ref** attribute, either on the forename or on the whole name component:

```
<forename nymRef="#N123" ref="#BLT">Tony</forename>
....

<person xml:id="BLT">
  <persName>Tony Blair</persName>
  <occupation>politician</occupation>
</person>
```

The <nym> element may be thought of as providing a specialised kind of dictionary entry. Like a dictionary entry, it may contain any element from the model.entryPart class, such as <form>, <etym>, etc. For example, we may show that the canonical form for a given nym has two orthographic variants in this way:

```
<nym xml:id="J451">
  <form>
    <orth xml:lang="en-US">Ian</orth>
    <orth xml:lang="en-x-Scots">Iain</orth>
  </form>
</nym>
```

Because a schema intending to make use of the <nym> or <listNym> element must include the dictionaries module as well as the namesdates module, many other elements are available in addition to <form>. For example, to provide a more complex etymological decomposition of a name, we might use the existing <etym> element, as follows:

```
<nym xml:id="XYZ">
  <form>Bogomil</form>
  <etym>Means <gloss>favoured by God</gloss> from the
  <lang>Slavic</lang> elements <mentioned xml:lang="ru">bog</mentioned>
    <gloss>God</gloss> and <mentioned xml:lang="ru">mil</mentioned>
    <gloss>favour</gloss>
  </etym>
</nym>
```

Where it is necessary to mark the substructure of nyms, this might be done by marking <seg> elements within the <form>:

```
<nym xml:id="ABC">
  <form>
    <choice>
      <seg type="morph">
        <seg>Bog</seg>
        <seg>o</seg>
        <seg>mil</seg>
      </seg>
      <seg type="morph">
        <seg>Bogo</seg>
        <seg>mil</seg>
      </seg>
    </choice>
  </form>
</nym>
```

The <seg> element used here is provided by the TEI linking module, which would therefore also need to be included in a schema built to validate such markup. Other possibilities for more detailed linguistic analysis are provided by elements included in that and the analysis (see *17. Simple Analytic Mechanisms*) or ISOfs modules (see *18. Feature Structures*).

Alternatively, each of the constituents of *Bogomil* might be regarded as a nym in its own right:

```
<nym xml:id="B1" type="part">
  <form>bog</form>
</nym>
<nym xml:id="M1" type="part">
  <form>mil</form>
</nym>
```

Within running text, a name can specify all the nyms associated with it:

```
...<name nymRef="#B1 #M1">Bogomul</name>...
```

Similarly, within a nym, the attribute **parts** is used to indicate its constituent parts, where these have been identified as distinct nyms:

```
<nym xml:id="BM1" parts="#B1 #M1">
  <form>Bogomil</form>
</nym>
```

The <nym> element may also combine a number of other <nym> elements together, where it is intended to show that they are all regarded as variations on the same root. Thus the different forms of the name John, all being derived from the same Latin root, may be represented as a hierarchic structure like this:

```
<nym xml:id="J45">
  <form xml:lang="la">Iohannes</form>
  <nym xml:id="J450">
    <form xml:lang="en">John</form>
    <nym xml:id="J4501">
      <form>Johnny</form>
    </nym>
    <nym xml:id="J4502">
      <form>Jon</form>
    </nym>
  </nym>
  <nym xml:id="J455">
    <form xml:lang="ru">Ivan</form>
  </nym>
  <nym xml:id="J453">
    <form xml:lang="fr">Jean</form>
  </nym>
</nym>
```

The <nym> element may be used for components of geographical or organizational names as well. For example:

```
<geogName key="LAEI1" type="hill">
  <geogFeat xml:lang="gd" nymRef="#LAIRG">Lairig</geogFeat>
  <name>Eilde</name>
</geogName>
...

<nym xml:id="LAIRG">
  <form xml:lang="gd">lairig</form>
  <def>sloping hill face</def>
</nym>
...
```

As noted above, use of these elements implies that both the dictionaries and the namesdates modules are included in a schema.

13.3.6 Dates and Times

The following elements for the encoding of dates and times were introduced in section *3.5.4. Dates and Times*:

<date> contains a date in any format.

<time> contains a phrase defining a time of day in any format.

The current module namesdates provides a mechanism for more detailed encoding of relative dates and times. A relative temporal expression describes a date or time with reference to some other (absolute) temporal expression, and thus may contain an <offset> element in addition to one or more <date> or <time> elements:

<offset> that part of a relative temporal or spatial expression which indicates the direction of the offset between the two place names, dates, or times involved in the expression.

As members of the att.datable and att.duration classes, which in turn are members of att.datable.w3c and att.duration.w3c respectively, the <date> and <time> elements share the following attributes:

att.datable.w3c provides attributes for normalization of elements that contain datable events using the W3C datatypes.

@**when** supplies the value of a date or time in a standard form, e.g. yyyy-mm-dd.

att.duration.w3c attributes for recording normalized temporal durations.

@**dur** (duration) indicates the length of this element in time.

13.3.6.1 Relative Dates and Times

As noted above, relative dates and times such as 'in the Two Hundredth and First Year of the Republic', 'twenty minutes before noon', and, more ambiguously, 'after the lamented death of the Doctor' or 'an hour after the game' have two distinct components. As well as the absolute temporal expression or event to which reference is made (e.g. 'noon', 'the game', 'the death of the Doctor', '[the foundation of] the Republic'), they also contain a description of the 'distance' between the time or date which is indicated and the referent expression (e.g. 'the Two Hundredth and First Year', 'twenty minutes', 'an hour'); and (optionally) an 'offset' describing the direction of the distance between the time or date indicated and the referent expression (e.g. 'of' implying after, 'before', 'after').

The 'distance' component of a relative temporal expression may be encoded as a temporal element in its own right using either <date> or <time>, or with the more generic <measure> element. A special element, <offset>, is provided by this module for encoding the 'offset' component of a relative temporal expression. The absolute temporal expression contained within the relative expression may be encoded with a <date> or <time> element; in turn, those elements may of course be relative, and thus contain <date> or <time> elements within themselves. This allows for deeply nested structures such as 'the third Sunday after the first Monday before Lammastide in the fifth year of the King's second marriage ... ' but so does natural language.

In the following examples, the **when** and **dur** attributes have been used to simplify processing of variant forms of expression:

```
<date when="1786-12-11">
  <date dur="P14D">A fortnight</date>
  <offset>before</offset>
  <date when="1786-12-25" type="holiday">Christmas 1786</date>
</date>
```

```
I reached the station <time when="14:15:00">
  <time dur="PT30M0S">precisely half an hour</time>
  <offset>after</offset>
  <time when="13:45:00" type="occasion">the departure of the afternoon train to Boston</time>
</time>
```

In the following example, a nested <date> element is used to show that 'my birthday' and the cited date are parts of the same temporal expression, and hence to disambiguate the phrase 'A week before my birthday on 9th December':

```
<date when="--12-02">
  <date>A week</date>
  <offset>before</offset>
  <date when="--12-09">
    <date type="occasion">my birthday</date>
     on <date>9th December</date>
  </date>
</date>
```

The alternative reading of this phrase could be encoded as follows:

```
<date when="--12-09">
  <date>A week</date>
  <offset>before</offset>
  <date type="occasion" when="--12-16">my birthday</date>
on <date>9th December</date>
</date>
```

Where more complex or ambiguous expressions are involved, and where it is desirable to make more explicit the interpretive processes required, the feature structure notation described in chapter *18. Feature Structures* is recommended. Consider, for example, the following temporal expression which occurs in the *Scottish Temperance Review* of August 1850, referring to the summer holiday known in Glasgow simply as 'the Fair':

```
Not only is the city,
<date ana="#gf50">during the Fair</date>, a horrible nucleus of
immorality and wickedness; it sends our multitudes to pollute and
demoralize the country.
```

For the definition of the **ana** attribute, see chapter *17. Simple Analytic Mechanisms* (in particular *17.2. Global Attributes for Simple Analyses*). It is used here to link the temporal phrase with an interpretation of it. Like most traditional fairs and market days, the Glasgow Fair was established by local custom and could vary from year to year. Consequently, in order to provide such an interpretation, it is necessary to drawn upon additional information which may or may not be located in the particular text in question. In this case, it is necessary at least to know the spatial and temporal context (year and place) of the fair referred to. These and other features required for the analysis of this particular temporal expression may be combined together as one feature structure of type **date-analysis**:

```
<fs xml:id="gf50" type="date-analysis">
  <f name="event">
    <string>the Fair</string>
  </f>
  <f name="place">
    <string>Glasgow</string>
  </f>
  <f name="year">
    <numeric value="1850"/>
  </f>
  <f name="from-value">
    <string>1850-08-08</string>
  </f>
  <f name="to-value">
    <string>1850-09-19</string>
  </f>
</fs>
```

For further discussion of feature structure representation see chapter *18. Feature Structures*.

13.3.6.2 Absolute Dates and Times

The following are examples of absolute temporal expressions.

```
The university's view
of American affairs produced a stinging attack by Edmund Burke in the
Commons debate of <date  when="1775-10-26">26 October 1775</date>
```

Source: [191]

```
<date  when="1993-05-14">Friday, 14 May 1993</date>
```

Source: [194]

It may be useful to categorize a temporal expression which is given in terms of a named event, such as a public holiday for dates, or a named time such as 'tea time' or 'matins':

```
In New York,
<date  type="occasion"  when="--01-01">New Years Day</date> is the
quietest of holidays, <date  when="--07-04"  type="occasion">Independence
Day</date> the most turbulent.
```

Absolute temporal expressions denoting times which are given in terms of seconds, minutes, hours, or of well-defined events (e.g. 'noon', 'sunset') may similarly be represented using the <time> element.

```
The train leaves for Boston at
<time  type="twentyfourHour"  when="13:45:00">13:45</time>
```

```
At <time  type="occasion">sunset</time> we walked to the beach.
```

```
The train leaves for Boston at
<time  xml:lang="en-US"  type="descriptive"  when="13:45:00-05:00"> a quarter of two
</time>
```

13.3.6.3 More Expressive Normalizations

The attributes for normalization of dates and times so far described use a standard format defined by *XML Schema Part 2: Datatypes Second Edition*. This format is widely accepted and has significant software support. It is essentially a profile of ISO 8601 *Data elements and interchange formats — Information interchange — Representation of dates and times*. The ISO standard provides formats not available in the W3C recommendation. For example, the capability to refer to a date by its ordinal date or week date, or to a calendar week. In cases where it is desirable to use these more specialized formats, this module provides a corresponding additional class of attributes for them:

att.datable.iso provides attributes for normalization of elements that contain datable events using the ISO 8601 standard.

- @**when-iso** supplies the value of a date or time in a standard form.
- @**notBefore-iso** specifies the earliest possible date for the event in standard form, e.g. yyyy-mm-dd.
- @**notAfter-iso** specifies the latest possible date for the event in standard form, e.g. yyyy-mm-dd.

@**from-iso** indicates the starting point of the period in standard form.

@**to-iso** indicates the ending point of the period in standard form.

att.duration.iso attributes for recording normalized temporal durations.

@**dur-iso** (duration) indicates the length of this element in time.

These attributes rely on the following datatype macros:

data.temporal.iso defines the range of attribute values expressing a temporal expression such as a date, a time, or a combination of them, that conform to the international standard *Data elements and interchange formats – Information interchange – Representation of dates and times.*

data.duration.iso defines the range of attribute values available for representation of a duration in time using ISO 8601 standard formats

The **when** and **dur** attributes are both used to provide a standardized or regularized form for the content of an element, conforming to a subset of the possible formats defined by the relevant international standard (ISO 8601) as profiled by *XML Schema Part 2: Datatypes Second Edition*.

For example:

```
<date when="1807-06-09">June 9th</date> The
period is approaching which will terminate my present
copartnership. On the <date when="1808-01-01">1st Jany.</date> next,
it expires by its own limitation.
```

Source: [48]

13.4 Module for Names and Dates

The module described in this chapter makes available the following components:

Module namesdates: Names and dates

- Elements defined: addName affiliation age birth bloc climate country death district education event faith floruit forename genName geo geogFeat geogName langKnowledge langKnown listNym listOrg listPerson listPlace location nameLink nationality nym occupation offset org orgName persName person personGrp place placeName population region relation relationGrp residence roleName settlement sex socecStatus state surname terrain trait
- Classes defined: att.datable.iso att.duration.iso model.persNamePart

The selection and combination of modules to form a TEI schema is described in *1.2. Defining a TEI Schema*.

Chapter 14

Tables, Formulæ, and Graphics

Many documents, both historical and contemporary, include not only text, but also graphics, artwork, and other images. Although some types of images can be represented directly with markup, it is more common practice to include such information by using a reference to an external entity (typically a URL) encoded in a suitable graphical notation.

In addition to graphic images, documents often contain material presented in graphical or tabular format. In such materials, details of layout and presentation may also be of comparatively greater significance or complexity than they are for running text. Indeed, it may often be difficult to make a clear distinction between details relating purely to the rendition of information and those relating to the information itself.

Finally, documents may contain mathematical formulæ or expressions in other formulaic notations, for which no notation is defined in these Guidelines.

These areas (graphics, tabular material, and mathematical or other formulæ) have in common that they have received considerable attention from many other standards bodies or similar professional groups. In part because of this, they may frequently be most conveniently encoded and processed using some notation not defined by these Guidelines. For these reasons, and others, we consider tables, formulæ, and graphics together in this chapter.

As with text markup in general, many incompatible formats have been proposed for the representation of graphics, formulæ, and tables in electronic form. Unfortunately, no single format as effective as XML in the domain of text has yet emerged for their interchange, to some extent because of the difficulty of representing the information these data formats convey independently of the way it is rendered.

The module defined by this chapter defines special purpose 'container' elements that can be used to encapsulate occurrences of such data within a TEI-conformant document in a portable way. Specific recommendations for the encoding of tables are provided in section *14.1. Tables* and recommendations for mathematical or other formulæ in section *14.2. Formulæ and Mathematical Expressions*. Specific recommendations for the encoding of graphic figures may be found in section *14.3. Specific Elements for Graphic Images*. The rest of the chapter is devoted to general problems of encoding graphic information.

There is at the time of writing no consensus on formats for graphical images, and such formats vary in many ways. We therefore provide (in section *14.4. Overview of Basic Graphics Concepts*) a brief discussion of the ways in which images may be represented, and (in section *14.5. Graphic Image Formats*) a list of formal names for those representations most popular at this time. Each one includes a very brief description. These Guidelines recommend a few particular representations as being the most widely supported and understood.

14.1 Tables

A table is the least 'graphic' of the elements discussed in this chapter. Almost any text structure can be presented as a series of rows and columns: one might, for example, choose to show a glossary or other form of list

in tabular form, without necessarily regarding it as a table. In such cases, the global **rend** attribute is an appropriate way of indicating that some element is being presented in tabular format, for example by using an appropriate display property in CSS. When tabular presentation is regarded as of less intrinsic importance, it is correspondingly simpler to encode descriptive or functional information about the contents of the table, for example to identify one cell as containing a name and another as containing a date, though the two methods may be combined.

When, however, particular elements are required to encode the tabular arrangement itself, then one or other of the various 'table schemas' now available may be preferable. The schemas in common use generally view a table as a special text element, made up of row elements, themselves composed of cells. Table cells generally appear in row-major order, with the first row from left to right, then the second row, and so on. Details of appearance such as column widths, border lines, and alignment are generally encoded by numerous attributes. Beyond this, however, such schemas differ greatly. This section begins by describing a table schema of this kind; a brief summary of some other widely available table schemas is also provided in section *14.1.2. Other Table Schemas*.

14.1.1 TEI Tables

For encoding tables of low to moderate complexity, these Guidelines provide the following special purpose elements:

<table> contains text displayed in tabular form, in rows and columns.

@**rows** indicates the number of rows in the table.

@**cols** (columns) indicates the number of columns in each row of the table.

<row> contains one row of a table.

<cell> contains one cell of a table.

The <table> element is defined as a member of the class inter; it may therefore appear both within other components (such as paragraphs), or between them, provided that the module defined in this chapter has been enabled, as described at the beginning of this chapter.

It is to a large extent arbitrary whether a table should be regarded as a series of rows or as a series of columns. For compatibility with currently available systems, however, these Guidelines require a row-by-row description of a table. It is also possible to describe a table simply as a series of cells; this may be useful for tabular material which is not presented as a simple matrix.

The attributes **rows** and **cols** may be used to indicate the size of a table, or to indicate that a particular cell or row of a table spans more than one row or column. For both tables and cells, rows and columns are always given in top-to-bottom, left-to-right order, although formatting properties such as those provided by CSS may be used to specify that they should be displayed differently. These Guidelines do not require that the size of a table be specified; for most formatting and many other applications, it will be necessary to process the whole table in two passes in any case.

Where cells span more than one column or row, the encoder must determine whether this is a purely presentational effect (in which case the **rend** attribute may be more appropriate), whether the part of the table affected would be better treated as a nested table, or whether to use the spanning attributes listed above.

The **role** attribute may be used to categorize a single cell, or set a default for all the cells in a given row. The present Guidelines distinguish the roles of label and data only, but the encoder may define other roles, such as 'derived', 'numeric', etc., as appropriate.

These three attributes are provided by the attribute class att.tableDecoration of which both <cell> and <row> are members; see further *1.3.1. Attribute Classes*.

The following simple example demonstrates how the data presented as a labelled list in section *3.7. Lists* might be represented by an encoder wishing to preserve its original appearance as a table:

```
<table  rend="boxed"  rows="2"  cols="2">
  <head  rend="it">Report of the conduct and progress
     of Ernest Pontifex.  Upper Vth form — half
     term ending Midsummer 1851</head>
  <row>
    <cell  role="label">Classics</cell>
    <cell>Idle listless and unimproving</cell>
  </row>
  <row>
    <cell  role="label">Mathematics</cell>
    <cell>ditto</cell>
  </row>
  <row>
    <cell  role="label">Divinity</cell>
    <cell>ditto</cell>
  </row>
  <row>
    <cell  role="label">Conduct in house</cell>
    <cell>Orderly</cell>
  </row>
  <row>
    <cell  role="label">General conduct</cell>
    <cell>Not satisfactory, on
        account of his great unpunctuality and inattention to
        duties</cell>
  </row>
</table>
```

Source: [25]

Note that this encoding makes no attempt to represent the full significance of the 'ditto' cells above; these might be regarded as simple links between the cells containing them and that to which they refer, or as virtual copies of it. For ways of representing either interpretation, see chapter *16. Linking, Segmentation, and Alignment*.

The following example demonstrates how a simple statistical table may be represented using this scheme:

```
<table  rows="4"  cols="4">
  <head>Poor Man's Lodgings in Norfolk (Mayhew, 1843)</head>
  <row  role="label">
    <cell/>
    <cell>Dossing Cribs or Lodging Houses</cell>
    <cell>Beds</cell>
    <cell>Needys or Nightly Lodgers</cell>
  </row>
  <row>
    <cell  role="label">Bury St Edmund's</cell>
    <cell>5</cell>
    <cell>8</cell>
    <cell>128</cell>
  </row>
  <row>
    <cell  role="label">Thetford</cell>
    <cell>3</cell>
    <cell>6</cell>
    <cell>36</cell>
  </row>
  <row>
```

```
    <cell  role="label">Attleboro'</cell>
    <cell>3</cell>
    <cell>5</cell>
    <cell>20</cell>
  </row>
  <row>
    <cell  role="label">Wymondham</cell>
    <cell>1</cell>
    <cell>11</cell>
    <cell>22</cell>
  </row>
</table>
```

Note the use of a blank cell in the first row to ensure that the column labels are correctly aligned with the data. Again, this encoding does not explicitly represent the alignment between column and row labels and the data to which they apply. Where the primary emphasis of an encoding is on the semantic content of a table, a more explicit mechanism for the representation of structured information such as that provided by the feature structure mechanism described in chapter *18. Feature Structures* may be preferred. Alternatively, the general purpose linkage and alignment mechanisms described in chapter *16. Linking, Segmentation, and Alignment* may also be applied to individual cells of a table.

The content of a table cell need not be simply character data. It may also contain any sequence of the phrase-level elements described in chapter *3. Elements Available in All TEI Documents*, thus allowing for the encoding of potentially more useful semantic information, as in the following example, where the fact that one cell contains a number and the other contains a place name has been explicitly recorded:

```
<table>
  <head>US State populations, 1990</head>
  <row>
    <cell>
      <name>Wyoming</name>
    </cell>
    <cell>
      <num>453,588</num>
    </cell>
  </row>
  <row>
    <cell>
      <name>Alaska</name>
    </cell>
    <cell>
      <num>550,043</num>
    </cell>
  </row>
  <row>
    <cell>
      <name>Montana</name>
    </cell>
    <cell>
      <num>799,065</num>
    </cell>
  </row>
  <row>
    <cell>
      <name>Rhode Island</name>
```

```
    </cell>
    <cell>
      <num>1,003,464</num>
    </cell>
  </row>
</table>
```

The **role** attribute provides a slightly less verbose means of conveying the same information:

```
<table>
  <head>US State populations, 1990</head>
  <row>
    <cell  role="statename">Wyoming </cell>
    <cell  role="pop">453,588 </cell>
  </row>
  <row>
    <cell  role="statename">Alaska </cell>
    <cell  role="pop">550,043 </cell>
  </row>
  <row>
    <cell  role="statename">Montana </cell>
    <cell  role="pop">799,065 </cell>
  </row>
  <row>
    <cell  role="statename">Rhode Island</cell>
    <cell  role="pop">1,003,464</cell>
  </row>
</table>
```

The use of semantically marked elements within a <cell> enables the encoder to convey something about the nature and significance of the information, rather than merely suggesting how to display it in rows and columns.

14.1.2 Other Table Schemas

Many authoring systems include built-in support for their own or for public table schemas. These provide an enhanced user interface and good formatting capabilities, but are often product-specific, despite their use of a XML markup language.

The DTD developed by the Association of American Publishers (AAP) and standardized in ANSI Z39.59 provided a very simple encoding for correspondingly simple tables. This has been further developed, together with the table DTD documented in ISO Technical Report 9537, and now forms part of ISO 12083. The TEI table model described above has functionality very similar to that defined by ISO 12083.

For more complex tables, the most effective publicly-available DTD is probably that developed by the US Department of Defense CALS project. This supports vertical and horizontal spanning and various kinds of text rotation and justification within cells and is also directly supported by a number of existing SGML software systems.

The CALS table model is much too complex to describe fully here; for historical background see `http://www.hbingham.com/technical/tables/calstbhs.htm`; for more recent simplifications of it and current implementations see `http://www.oasis-open/specs/tablemodels.php`. As with any other XML vocabulary, the XML version of the CALS model may readily be included in a TEI schema, using the techniques described in *23.2. Personalization and Customization*.

The XHTML table model (XHTML™ 1.0 The Extensible HyperText Markup Language (Second Edition) (2000)) is based on the HTML table model (Ragget et al. (eds.) (1999)). Both models support arrangement of arbitrary data into rows and columns of cells. Table rows and columns may be grouped to convey additional structural information and may be rendered by user agents in ways that emphasize this structure. Support for incremental rendering of tables and for rendering on 'non-visual' user agents is also available. Special elements and attributes are provided to associate metadata with tables. They indicate the table's purpose, or are for the benefit of people using speech or Braille-based user agents. Tables are not recommended for use purely as a means to lay out document content, as this leads to many accessibility problems (see further `http://www.w3.org/TR/WCAG10-HTML-TECHS/#tables`). Stylesheets provide a far more effective means of controlling layout and other visual characteristics in both HTML and XML documents.

14.2 Formulæ and Mathematical Expressions

Mathematical and chemical formulæ pose problems similar to those posed by tables in that rendition may be of great significance and hard to disentangle from content. They also require access to a wide range of special characters, for most of which standard entity names already exist in the documented ISO entity sets (see further chapters *vi Languages and Character Sets* and *5. Representation of Non-standard Characters and Glyphs*).

Formulæ and tables are also similar in that well-researched and detailed DTD fragments have already been developed for them independently of the TEI. They differ in that (for mathematics at least) there also exists a richly detailed text-based but non-SGML notation which is very widely used: this is the TeX system, and the sets of descriptive macros developed for it such as LaTeX, AMS-TeX, and AMS-LaTeX.

The AAP and ISO standards mentioned in section *14.1. Tables* above both provide DTDs for equations as well as for tables, which now form part of ISO 12083. The European Mathematical Trust, an organization set up specifically to enhance research support for European mathematicians, has also defined a general purpose mathematical DTD known as EuroMath (`http://www.dcs.fmph.uniba.sk/~emt/`), for which it provides both software and services.

Most if not all of the functionality provided by these DTDs can now be found in the OpenMath and MathML XML-based systems briefly described below.

As with tables, in all the SGML and XML solutions a tension exists between the need to encode the way a formula is written (its appearance) and the need to represent its semantics. If the object of the encoding is purely to act as an interchange format among different formatting programs, then there is no need to represent the mathematical meaning of an expression. If however the object is to use the encoding as input to an algebraic manipulation system (such as Mathematica or Maple) or a database system, clearly simply representing superscripts and subscripts will be inadequate.

The present Guidelines make no attempt to add to the number of available DTDs for representing formulæ. Instead, we recommend that the user make an informed choice from those already available. The module described in this chapter makes available only the following element, which should be used to encode any formula, no matter what notation is employed:

<formula> contains a mathematical or other formula.

@notation supplies the name of a previously defined notation used for the content of the element.

By default, a <formula> is assumed to contain character data which is not validated in any way:

```
<formula notation="TeX">$e=mc^2$</formula>
```

The character data must still be well-formed, of course, which means that < and & must be escaped with entity references or numeric character references, e.g.

```
<formula  notation="TeX">$\matrix{0 &amp; 1\cr&lt;0&amp;>1}$</formula>
```

If desired, the content of the <formula> element may be redefined to include elements defined by some other module, such as that of ISO 12083, or to use elements from the more recently defined OpenMath or MathML schemas.

When the content of a <formula> element is not expressed in XML the notation used should be specified using the **notation** attribute as above, and in the following longer example:

```
<p>Achilles runs ten times faster than the tortoise and
gives the animal a headstart of ten meters. Achilles runs
those ten meters, the tortoise one; Achilles runs that
meter, the tortoise runs a decimeter; Achilles runs that
decimeter, the tortoise runs a centimeter; Achilles runs
that centimeter, the tortoise, a millimeter; Fleet-footed
Achilles, the millimeter, the tortoise, a tenth of a
millimeter, and so on to infinity, without the tortoise ever
being overtaken. . . Such is the customary version.

<!-- ... -->
The problem does not change, as you can see; but I would
like to know the name of the poet who provided it with a
hero and a tortoise. To those magical competitors and to
the series
<formula  notation="TeX">$$
    {1 \over 10} +
    {1 \over 100} +
    {1 \over 1000} +
    {1 \over 10,\!000} +
    \dots
    $$</formula>
the argument owes its fame.</p>
```

Source: [19]

The **notation** attribute supplies the name of a notation ('TeX'), which is expected to be identified somewhere in document metadata.

Mathematical Markup Language (MathML) (Carlisle et al. (eds.) (2003)) is a vocabulary for describing mathematical notation, capturing both its structure and content. It provides two types of markup: Presentation Markup, which captures the **notational** structure of an expression and could be seen as the 'TeX for the Web' and Content Markup, which captures the **mathematical** structure of an expression. Most of its content elements correspond with the range of operators, relations, and named functions typically found at the high-school level of mathematics. The tortoise example given above in TeX can be re-expressed in MathML as

```
<m:math>
  <m:mfrac>
    <m:mrow>
      <m:mn>1</m:mn>
    </m:mrow>
    <m:mrow>
      <m:mn>10</m:mn>
    </m:mrow>
  </m:mfrac>
  <m:mo>+</m:mo>
```

```
  <m:mfrac>
    <m:mrow>
      <m:mn>1</m:mn>
    </m:mrow>
    <m:mrow>
      <m:mn>100</m:mn>
    </m:mrow>
  </m:mfrac>
  <m:mo>+</m:mo>
  <m:mfrac>
    <m:mrow>
      <m:mn>1</m:mn>
    </m:mrow>
    <m:mrow>
      <m:mn>1000</m:mn>
    </m:mrow>
  </m:mfrac>
  <m:mo>+</m:mo>
  <m:mfrac>
    <m:mrow>
      <m:mn>1</m:mn>
    </m:mrow>
    <m:mrow>
      <m:mn>10000</m:mn>
    </m:mrow>
  </m:mfrac>
  <m:mo>+</m:mo>
  <m:mo>…</m:mo>
</m:math>
```

MathML 2.0 provides support for a 'Semantic Math-Web', XML namespaces, and other current XML standards, such as XML DOM, OMG IDL, ECMAScript, and Java. It also provided a modularized version of the MathML DTD so that MathML fragments 'embedded' in XHTML 1.1 documents can be correctly validated.

The OpenMath (`http://www.nag.co.uk/projects/OpenMath.html`) project is coordinated by the OpenMath Society (`http://www.openmath.org/`) and funded by the European Commission under the Esprit Multimedia Standards Initiative that commenced in September 1997. It is likely to become a key standard for communicating semantically rich representations of mathematical objects both on and off the Web in a platform-independent manner.

The OpenMath Standard (`http://www.openmath.org/V2/standard/index.html`) consists of specifications for

1. OpenMath objects, representing the structure of formulæ (`http://www.openmath.org/V2/standard/objects.html`);

2. Content Dictionaries, providing semantic context (`http://www.openmath.org/V2/standard/cd.html`);

3. Encodings, both binary (`http://www.openmath.org/V2/standard/binary.html`) and XML (`http://www.openmath.org/V2/standard/xml.html`).

OpenMath and MathML have certain common aspects. They both use prefix operators, both are XML-based and they both construct their objects by applying certain rules recursively. Such similarities facilitate mapping between the two standards. There are also some key differences between MathML and OpenMath.

OpenMath does not provide support for presentation of mathematical objects and its scope of semantically-oriented elements is much broader that of MathML, with the expressive power to cover virtually all areas of computational mathematics. In fact, a particular set of Content Dictionaries, the 'MathML CD Group', covers the same areas of mathematics as the Content Markup elements of MathML 2.0.

Finally, OMDoc (`http://www.mathweb.org/omdoc/`) is an extension of the OpenMath standard that supplies markup for structures such as axioms, theorems, proofs, definitions, texts (mixing formal content with mathematical text).

In-line versus block placement for an equation can be distinguished if desired, via the global **rend** attribute. The global **n** and **xml:id** attributes may also be used to label or identify the formula, as in the following example:

```
<p>The volume of a
sphere is given by the formula:
<formula xml:id="f12" n="12" rend="inline">
   <m:math>
     <m:mi>V</m:mi>
     <m:mo>=</m:mo>
     <m:mfrac>
       <m:mrow>
         <m:mn>4</m:mn>
       </m:mrow>
       <m:mrow>
         <m:mn>3</m:mn>
       </m:mrow>
     </m:mfrac>
     <m:mi>π</m:mi>
     <m:msup>
       <m:mrow>
         <m:mi>r</m:mi>
       </m:mrow>
       <m:mrow>
         <m:mn>3</m:mn>
       </m:mrow>
     </m:msup>
   </m:math>
 </formula>
which is readily calculated.</p>
<p>As we have seen in equation
<ptr target="#f12"/>, ... </p>
```

14.3 Specific Elements for Graphic Images

The following special purpose elements are used to indicate the presence of graphic images within a document:

<figure> groups elements representing or containing graphic information such as an illustration or figure.

<graphic/> indicates the location of an inline graphic, illustration, or figure.

<binaryObject> provides encoded binary data representing an inline graphic or other object.

<figDesc> (description of figure) contains a brief prose description of the appearance or content of a graphic figure, for use when documenting an image without displaying it.

The <graphic> and <binaryObject> elements form part of the common core module, and are discussed in section *3.9. Graphics and other non-textual components.*

The <figure> element is used to contain images, captions, and textual descriptions of the pictures. The images themselves are specified using the <graphic> element, whose **url** attribute provides the location of an image. For example:

```
<figure>
  <graphic  url="Fig1.pdf"/>
</figure>
```

Three kinds of content may be supplied inside a <figure> element: the element <head> may be used to transcribe (or supply) a descriptive heading or title for the graphic itself as in this example:

```
<figure>
  <graphic  url="Fig1.pdf"/>
  <head>Figure One: The View from the
     Bridge</head>
</figure>
```

Figures are often accompanied not only by a title or heading, but by a paragraph or so of commentary or caption. One or more <p> or <ab> elements may be used to transcribe any caption or discussion of the figure in the source:

```
<figure>
  <graphic  url="pullman.png"/>
  <head>Above:</head>
  <p>The drawing room of the Pullman house, the white and gold saloon
     where the magnate delighted in giving receptions for several
     hundred people.</p>
  <figDesc>The figure shows an elaborately decorated room, at least
     twenty-five feet side to side and fifty feet long, with ornate
     mouldings and Corinthian columns on the walls, overstuffed
     armchairs and loveseats arranged in several conversational
     groupings, and two large chandeliers.</figDesc>
</figure>
```

Source: [131]

Here, the paragraph 'The drawing room ... several hundred people' is transcribed from the source, while the description is provided by the encoder, for use by applications which cannot display the graphic directly. In documents created in electronic form with the needs of print-handicapped readers in mind, the <figDesc> element may be provided by the author rather than a subsequent encoder.

```
<figure>
  <graphic  url="Fig1.jpg"/>
  <head>Figure One: The View from the Bridge</head>
  <figDesc>A Whistleresque view showing four
     or five sailing boats in the foreground, and a
     series of buoys strung out between them.</figDesc>
</figure>
```

Where the graphic itself contains large amounts of text, perhaps with a complex structure, and perhaps difficult to distinguish from the graphic, the encoder should choose whether to regard the graphic as containing the text (in which case, a nested <floatingText> element may be included within the <figure> element) or to regard the enclosed text as being a separate division of the <text> element in which the graphic appears. In this latter case, an appropriate or <div1> (etc.) element may be used for the text represented within

the graphic, and the <figure> element embedded within it. The choice will depend to a large degree on the encoder's understanding of the relationship between the graphic and the surrounding text.

A figure which is internally divided, or contains sub-figures, may be encoded with nested <figure> elements, as in the following example.

```
<figure n="6.45">
  <figure n="a">
    <graphic url="./figs/6.45a.png"/>
    <ab type="caption">Parallel</ab>
  </figure>
  <figure n="b">
    <graphic url="./figs/6.45b.png"/>
    <ab type="caption">Perspective</ab>
  </figure>
  <ab type="caption">The two canonical view volumes, for the (a) parallel
     and (b) perspective projections. Note that -z is to the right.</ab>
</figure>
```

Source: [75]

Like any other element in the TEI scheme, figures may be given identifiers so that they can be aligned with other elements, and linked to or from them, as described in chapter *16. Linking, Segmentation, and Alignment*. Some common examples are discussed briefly here; full information is provided in that chapter.

It is often desirable to maintain two versions of an image in an electronic file: one a low resolution or 'thumbnail' version which, when selected by the user, causes the other, high resolution, version to be accessed. In TEI terms, the thumbnail image acts as a reference to the other. Supposing that a thumbnail version of the figure discussed above is available as fig1th.png", we might embed a reference to the image using the simple <ref> element discussed in section *3.6. Simple Links and Cross-References*:

```
<ref target="#IM1">Click here
<graphic url="fig1th.png"/>
for enlightenment
</ref>
<figure xml:id="IM1">
  <graphic url="fig1.jpg"/>
</figure>
```

Another common requirement is to associate part or the whole of an image with a textual element not necessarily contiguous to it in the text; this is sometimes known as a callout. When the module for transcription is included in a schema, specific attributes for parts of a text and parts (or all) of a digital image are available; these are discussed in *11.1. Digital Facsimiles*. In addition, chapter *16. Linking, Segmentation, and Alignment* may be consulted for other mechanisms available for this purpose.

The following example assumes that we wish to associate one portion of the image held as 'fig1' with chapter two of some text, and another portion of it with chapter three. The application may be thought of as a hypertext browser in which the user selects from a graphic image which part of a text to read next, but the mechanism is independent of this particular application.

The first requirement is some way of identifying and hence pointing to sub-parts of a graphic image. This may be done by pointing into an XML graphic representation, for example an SVG file. Thus

```
<ptr xml:id="PD1" target="Fig1.svg#object1"/>
<ptr xml:id="PD2" target="Fig1.svg#object2"/>
```

These <ptr> elements identify two areas within the image 'Fig1' by pointing at elements inside the XML file Fig1.svg, which contains the following.

```
<svg xmlns="http://www.w3.org/2000/svg"
  width="8cm"  height="3cm"  viewBox="2 1 8 3">
  <g xmlns="http://www.w3.org/2000/svg"
    id="object1">
    <ellipse xmlns="http://www.w3.org/2000/svg"

       style="fill: #ffffff"
       cx="3.875"
       cy="3.025"
       rx="1.175"
       ry="1.175"/>
  </g>
  <g xmlns="http://www.w3.org/2000/svg"
    id="object2">
    <rect xmlns="http://www.w3.org/2000/svg"

       style="fill: #a81616"
       x="7.8"
       y="1.9"
       width="2.17581"
       height="2.24833"/>
  </g>
</svg>
```

The next requirement is some way of identifying the parts of the document to which a link is to be made. The most obvious way of doing this is to use the global **xml:id** attribute:

```
<div1  type="chapter"  xml:id="CHAP1">
<!-- ... -->
</div1>
<div1  type="chapter"  xml:id="CHAP2">
<!-- ... -->
</div1>
```

Now, all that is needed to linking these areas to the relevant chapters is a <linkGrp> element, as described in section *16.1. Links*:

```
<linkGrp  type="callout">
  <link  targets="#CHAP1 #PD1"/>
  <link  targets="#CHAP2 #PD2"/>
</linkGrp>
```

In this example, the SVG representation of the graphic is stored externally to the TEI document and linked by means of a pointer. It is also possible to embed the SVG representation directly within the TEI by extending the content model of the <figure> element to permit an element <svg> from the SVG namespace. Like other customizations of the TEI scheme, this is carried out using the techniques documented in section *1.2. Defining a TEI Schema*; further examples are provided in chapter *16. Linking, Segmentation, and Alignment*.

14.4 Overview of Basic Graphics Concepts

The first major distinction in graphic representation is that between raster graphics and vector graphics. A raster image is a list of points, or dots. Scanners, fax machines and other simple devices easily produce digital raster images, and such images are therefore quite common. A vector image, in contrast, is a list of geometrical objects, such as lines, circles, arcs, or even cubes. These are much more difficult to produce, and so are mainly encountered as the output of sophisticated systems such as architectural and engineering CAD programs.

Raster images are difficult to modify because by definition they only encode single points: a line, for example, cannot grow or shrink as such, since it is not identified as such. Only its component parts are identified, and only they can be manipulated. Therefore the resolution or dot-size of a raster image is important, which is not the case with vector images. It is also far more difficult to convert raster images to vector images than to perform the opposite conversion. Raster images generally require more storage space than vector images, and a wide variety of methods exists for compressing them; the variation in these methods leads to corresponding variations in representations for storage and transmission of raster images.

Motion video usually consists of a long series of raster images. Data compression is even more effective on video than on single raster images (mainly owing to redundancy which arises from the usual similarity of adjacent frames). Notations for representing full-motion video are hotly debated at this time, and any user of these Guidelines would do well to obtain up-to-date expert advice before undertaking a project using them.

The compression methods used with any of these image types may be 'lossy' or 'lossless'. Methods for lossy compression save space by discarding a small portion of the image's detail, such as fine distinctions of shading. When decompressed, therefore, such an image will be only a close approximation of the original. In contrast, lossless compression guarantees that the exact uncompressed image will be reproducible from the compressed form: only truly redundant information is removed. In general, therefore, lossless compression does not save quite so much space as lossy compression, though it does guarantee fidelity to the original uncompressed image.

Raster images may be characterized by their resolution, which is the number of dots per inch used to represent the image. Doubling the resolution will give a more precise image, but also quadruple the storage requirement (before compression), and affect processing time for any operations to be performed, such as displaying an image for a reader. Motion video also has resolution in time: the number of frames to be shown per second. Encoders should consider carefully what resolution(s) and frame rate(s) to use for particular applications; these Guidelines express no recommendation in this matter, save the universal ones of consistency and documentation.

Within any image, it is typical to refer to locations via Cartesian coordinate axes: values for x, y, and sometimes z and/or time. However, graphic notations vary in whether coordinates count from left-to-right and top-to-bottom, or another way. They also vary in whether coordinates are considered real (inches, millimeters, and so on), or virtual (dots). These Guidelines do not recommend any of these methods over another, but all decisions made should be applied consistently, and documented in the <encodingDesc> section of the TEI header.[1]

Methods of aligning images and text are discussed in *11.1. Digital Facsimiles*.

The chromatic values of an image may be rendered in many different ways. In monochrome images every displayed point is either black or white. In gray-scale images, each point is rendered in some shade of gray, the number of shades varying from system to system. In true polychrome images, points are rendered in different hues, again with varying limitations affecting the number of distinct shades and the means by which they are displayed.

[1] Since no special purpose element is provided for this purpose by the current version of the Guidelines, such information should be provided as one or more distinct paragraphs at the end of the <encodingDesc> element described in section *2.3. The Encoding Description*.

14.5 Graphic Image Formats

As noted above, there exists a wide variety of different graphics formats, and the following list is in no way exhaustive. Moreover, inclusion of any format in this list should not be taken as indicating endorsement by the TEI of this format or any products associated with it. Some of the formats listed here are proprietary to a greater or lesser extent and cannot therefore be regarded as standards in any meaningful sense. They are however widely used by many different vendors.

The following formats are widely used at the present time, and likely to remain supported by more than one vendor's software:

- BMP: Microsoft bitmap format
- CGM: Computer Graphics Metafile
- GIF: Graphics Interchange Format
- JPEG: Joint Photographic Expert Group
- PBM: Portable Bit Map
- PCX: IBM PC raster format
- PICT: Macintosh drawing format
- PNG: Portable Network Graphics format
- Photo-CD: Kodak Photo Compact Disk format
- QuickTime: Apple real-time image system
- SMIL: Synchronized Multimedia Integration Language format
- SVG: Scalable Vector Graphics format
- TIFF: Tagged Image File Format

Brief descriptions of all the above are given below. Where possible, current addresses or other contact information are shown for the originator of each format. Many formal standards, especially those promulgated by ISO and many related national organizations (ANSI, DIN, BSI, and many more), are available from those national organizations. Addresses may be found in any standard organizational directory for the country in question.

14.5.1 Vector Graphic Formats

CGM: Computer Graphics Metafile This vector graphics format is specified by an ISO standard, ISO 8632:1987, amended in 1990. It defines binary, character, and plain-text encodings; the non-binary forms are safer for blind interchange, especially over networks. Documentation on CGM is available from ISO and from its member national bodies such as AFNOR, ANSI, BSI, DIN, JIS, etc.

SVG: Scalable Vector Graphics format SVG is a language for describing two-dimensional vector and mixed vector or raster graphics in XML. It is defined by the Scalable Vector Graphics (SVG) 1.0 Specification, W3C Recommendation, 04 September 2001, and is available at `http://www.w3.org/TR/2001/REC-SVG-20010904/`.

PICT: Macintosh drawing format This format is universally supported on Macintosh (tm) systems, and readable by a limited range of software for other systems. Documentation is available from Apple Computer Company, Cupertino, California USA.

14.5.2 Raster Graphic Formats

PNG: Portable Network Graphics format PNG is a non-proprietary raster format currently widely available. It provides an extensible file format for the lossless, portable, well-compressed storage of raster images.

Indexed-color, grayscale, and truecolor images are supported, plus an optional alpha channel. Sample depths range from 1 to 16 bits. It is defined by IETF RFC 2083, March 1997.

TIFF: Tagged Image File Format Currently the most widely supported raster image format, especially for black and white images, TIFF is also one of the few formats commonly supported on more than one operating system. The drawback to TIFF is that it actually is a wrapper for several formats, and some TIFF-supporting software does not support all variants. TIFF files may use LZW, CCITT Group 4, or PackBits compression methods, or may use no compression at all. Also, TIFF files may be monochrome, greyscale, or polychromatic. All such options should be specified in prose at the end of the <encodingDesc> section of the TEI header for any document including TIFF images. TIFF is owned by Aldus Corporation. Documentation on TIFF is available from them at Craigcook Castle, Craigcook Road, Edinburgh EH4 3UH, Scotland, or 411 First Avenue South, Seattle, Washington 98104 USA.

GIF: Graphics Interchange Format Raster images are widely available in this form, which was created by CompuServe Information Services, but has by now been implemented for many other systems as well. Documentation on GIF is copyright by, and is available from, CompuServe Incorporated, Graphics Technology Department, 5000 Arlington Center Boulevard, Columbus, Ohio 43220 USA.

PBM: Portable Bit Map PBM files are easy to process, eschewing all compression in favor of transparency of file format. PBM files can, of course, be compressed by generic file-compression tools for storage and transfer. Public domain software exists which will convert many other formats to and from PBM. Documentation on PBM is copyright by Jeff Poskanzer, and is available widely on the Internet.

PCX: IBM PC raster format This format is used by most IBM PC paint programs, and supports both monochrome and polychromatic images. Documentation is available from ZSoft Corporation, Technical Support Department, ATTN: Technical Reference Manual, 450 Franklin Rd. Suite 100, Marietta, GA 30067 USA.

BMP: Microsoft bitmap format This format is the standard raster format for computer using Microsoft Windows (tm) or Presentation Manager (tm). Documentation is available from Microsoft Corporation.

14.5.3 Photographic and Motion Video Formats

JPEG: Joint Photographic Experts Group This standard is sponsored by CCITT and by ISO. It is ISO/IEC Draft International Standard 10918-1, and CCITT T.81. It handles monochrome and polychromatic images with a variety of compression techniques. JPEG per se, like CCITT Group IV, must be encapsulated before transmission; this can be done via TIFF, or via the JPEG File Interchange Format (JFIF), as commonly done for Internet delivery.

QuickTime: Apple real-time image system QuickTime is a proprietary method introduced by Apple Computer Company to synchronize the display of various data. The data can include frames of video, sound, lighting control mechanisms, and other things. Viewers for QuickTime productions are available for Apple and other computers. Further information is available from Apple Computer Incorporated, 10201 North de Anza Boulevard MS 23AQ, Cupertino, California 95014 USA.

Photo-CD: Kodak Photo Compact Disk format This format was introduced by Kodak for rasterizing photographs and storing them on CD-ROMs (about one hundred 35mm file images fit on one disk), for display on televisions or CD-I systems. Information on Photo-CD is available from Kodak Limited, Research and Development, Headstone Drive, Harrow, Middlesex HA1 4TY, UK.

SMIL: Synchronized Multimedia Integration Language format SMIL is a W3C Recommendation which supports the integration of independent multimedia objects into a synchronized multimedia presentation. It provides multimedia authors with easily-defined basic timing relationships, fine-tuned synchronization, spatial layout, direct inclusion of non-text and non-image media objects, hyperlink support for time-based media, adaptiveness to varying user and system characteristics. SMIL 1.0 (`http://www.w3.org/TR/REC-smil/`) became a W3C Recommendation on June 15, 1998, and was further developed in SMIL 2.0. SMIL 2.0 adds native support for transitions, animation, event-based interaction, extended layout facilities, and more sophisticated timing and synchronization primitives to the SMIL 1.0 language. It also allows reuse of SMIL syntax and semantics in other XML-based languages, in particular those who need to represent timing and synchronization. For example, SMIL 2.0 components are used for integrating timing into XHTML Document Types and into SVG. SMIL 2.0 also provides recommendations for Document Types based on SMIL 2.0 Modules (`http://www.w3.org/TR/smil20/smil-modules.html`). One such Document Type is the SMIL 2.0 Language Profile (`http://www.w3.org/TR/smil20/smil20-profile.html`). It contains support for all of the major SMIL 2.0 features including animation, content control, layout, linking, media object, meta-information, structure, timing, and transition effects and is designed for Web clients that support direct playback from SMIL 2.0 markup. SMIL 2.0 (`http://www.w3.org/TR/smil20/`) became a W3C Recommendation on August 7, 2001, becoming the first vocabulary to provide XML Schema support and to have reached such status.

As noted above, the reader will encounter many, many other graphics formats.

14.6 Module for Tables, Formulæ, and Graphics

The module described in this chapter provides the following features:

Module figures: Tables, formulæ, and figures

- Elements defined: cell figDesc figure formula row table

The selection and combination of modules to form a TEI schema is described in *1.2. Defining a TEI Schema*.

Chapter 15

Language Corpora

The term language corpus is used to mean a number of rather different things. It may refer simply to any collection of linguistic data (for example, written, spoken, signed, or multimodal), although many practitioners prefer to reserve it for collections which have been organized or collected with a particular end in view, generally to characterize a particular state or variety of one or more languages. Because opinions as to the best method of achieving this goal differ, various subcategories of corpora have also been identified. For our purposes however, the distinguishing characteristic of a corpus is that its components have been selected or structured according to some conscious set of design criteria.

These design criteria may be very simple and undemanding, or very sophisticated. A corpus may be intended to represent (in the statistical sense) a particular linguistic variety or sublanguage, or it may be intended to represent all aspects of some assumed 'core' language. A corpus may be made up of whole texts or of fragments or text samples. It may be a 'closed' corpus, or an 'open' or 'monitor' corpus, the composition of which may change over time. However, since an open corpus is of necessity finite at any particular point in time, the only likely effect of its expansibility from the encoding point of view may be some increased difficulty in maintaining consistent encoding practices (see further section *15.5. Recommendations for the Encoding of Large Corpora*). For simplicity, therefore, our discussion largely concerns ways of encoding closed corpora, regarded as single but composite texts.

Language corpora are regarded by these Guidelines as composite texts rather than unitary texts (on this distinction, see chapter *4. Default Text Structure*). This is because although each discrete sample of language in a corpus clearly has a claim to be considered as a text in its own right, it is also regarded as a subdivision of some larger object, if only for convenience of analysis. Corpora share a number of characteristics with other types of composite texts, including anthologies and collections. Most notably, different components of composite texts may exhibit different structural properties (for example, some may be composed of verse, and others of prose), thus potentially requiring elements from different TEI modules.

Aside from these high-level structural differences, and possibly differences of scale, the encoding of language corpora and the encoding of individual texts present identical sets of problems. Any of the encoding techniques and elements presented in other chapters of these Guidelines may therefore prove relevant to some aspect of corpus encoding and may be used in corpora. Therefore, we do not repeat here the discusssion of such fundamental matters as the representation of multiple character sets (see chapter *vi Languages and Character Sets*); nor do we attempt to summarize the variety of elements provided for encoding basic structural features such as quoted or highlighted phrases, cross-references, lists, notes, editorial changes and reference systems (see chapter *3. Elements Available in All TEI Documents*). In addition to these general purpose elements, these Guidelines offer a range of more specialized sets of tags which may be of use in certain specialized corpora, for example those consisting primarily of verse (chapter *6. Verse*), drama (chapter *7. Performance Texts*), transcriptions of spoken text (chapter *8. Transcriptions of Speech*), etc. Chapter *1. The TEI Infrastructure*

should be reviewed for details of how these and other components of the Guidelines should be tailored to create a document type definition appropriate to a given application. In sum, it should not be asssumed that only the matters specifically addressed in this chapter are of importance for corpus creators.

This chapter does however include some other material relevant to corpora and corpus-building, for which no other location appeared suitable. It begins with a review of the distinction between unitary and composite texts, and of the different methods provided by these Guidelines for representing composite texts of different kinds (section *15.1. Varieties of Composite Text*). Section *15.2. Contextual Information* describes a set of additional header elements provided for the documentation of contextual information, of importance largely though not exclusively to language corpora. This is the additional module for language corpora proper. Section *15.3. Associating Contextual Information with a Text* discusses a mechanism by which individual parts of the TEI Header may be associated with different parts of a TEI-conformant text. Section *15.4. Linguistic Annotation of Corpora* reviews various methods of providing linguistic annotation in corpora, with some specific examples of relevance to current practice in corpus linguistics. Finally, section *15.5. Recommendations for the Encoding of Large Corpora* provides some general recommendations about the use of these Guidelines in the building of large corpora.

15.1 Varieties of Composite Text

Both unitary and composite texts may be encoded using these Guidelines; composite texts, including corpora, will typically make use of the following tags for their top-level organization.

<teiCorpus> contains the whole of a TEI encoded corpus, comprising a single corpus header and one or more TEI elements, each containing a single text header and a text.

<TEI> (TEI document) contains a single TEI-conformant document, comprising a TEI header and a text, either in isolation or as part of a <teiCorpus> element.

<teiHeader> (TEI Header) supplies the descriptive and declarative information making up an electronic title page prefixed to every TEI-conformant text.

@type specifies the kind of document to which the header is attached, for example whether it is a corpus or individual text.

<text> contains a single text of any kind, whether unitary or composite, for example a poem or drama, a collection of essays, a novel, a dictionary, or a corpus sample.

<group> contains the body of a composite text, grouping together a sequence of distinct texts (or groups of such texts) which are regarded as a unit for some purpose, for example the collected works of an author, a sequence of prose essays, etc.

Full descriptions of these may be found in chapter *2. The TEI Header* (for <teiHeader>), and chapter *4. Default Text Structure* (for <teiCorpus><TEI>, <text> and <group>); this section discusses their application to composite texts in particular.

In these Guidelines, the word text refers to any stretch of discourse, whether complete or incomplete, unitary or composite, which the encoder chooses (perhaps merely for purposes of analytic convenience) to regard as a unit. The term composite text refers to texts within which other texts appear; the following common cases may be distinguished:

- language corpora
- collections or anthologies
- poem cycles and epistolary works (novels or essays written in the form of collections or series of letters)
- otherwise unitary texts, within which one or more subordinate texts are embedded

The elements listed above may be combined to encode each of these varieties of composite text in different ways.

In corpora, the component samples are clearly distinct texts, but the systematic collection, standardized preparation, and common markup of the corpus often make it useful to treat the entire corpus as a unit, too. Some corpora may become so well established as to be regarded as texts in their own right; the Brown and LOB corpora are now close to achieving this status.

The <teiCorpus> element is intended for the encoding of language corpora, though it may also be useful in encoding newspapers, electronic anthologies, and other disparate collections of material. The individual samples in the corpus are encoded as separate <TEI> elements, and the entire corpus is enclosed in a <teiCorpus> element. Each sample has the usual structure for a <TEI> document, comprising a <teiHeader> followed by a <text> element. The corpus, too, has a corpus-level <teiHeader> element, in which the corpus as a whole, and encoding practices common to multiple samples may be described. The overall structure of a TEI-conformant corpus is thus:

```
<teiCorpus>
  <teiHeader type="corpus"/>
  <TEI>
    <teiHeader type="text"/>
    <text/>
  </TEI>
  <TEI>
    <teiHeader type="text"/>
    <text/>
  </TEI>
</teiCorpus>
```

Header information which relates to the whole corpus rather than to individual components of it should be factored out and included in the <teiHeader> element prefixed to the whole. This two-level structure allows for contextual information to be specified at the corpus level, at the individual text level, or at both. Discussion of the kinds of information which may thus be specified is provided below, in section *15.2. Contextual Information*, as well as in chapter *2. The TEI Header*. Information of this type should in general be specified only once: a variety of methods are provided for associating it with individual components of a corpus, as further described in section *15.3. Associating Contextual Information with a Text*.

In some cases, the design of a corpus is reflected in its internal structure. For example, a corpus of newspaper extracts might be arranged to combine all stories of one type (reportage, editorial, reviews, etc.) into some higher-level grouping, possibly with sub-groups for date, region, etc. The <teiCorpus> element provides no direct support for reflecting such internal corpus structure in the markup: it treats the corpus as an undifferentiated series of components, each tagged <TEI>.

If it is essential to reflect a single permanent organization of a corpus into sub- and sub-sub-corpora, then the corpus or the high-level subcorpora may be encoded as composite texts, using the <group> element described below and in section *4.3.1. Grouped Texts*. The mechanisms for corpus characterization described in this chapter, however, are designed to reduce the need to do this. Useful groupings of components may easily be expressed using the text classification and identification elements described in section *15.2.1. The Text Description*, and those for associating declarations with corpus components described in section *15.3. Associating Contextual Information with a Text*. These methods also allow several different methods of text grouping to co-exist, each to be used as needed at different times. This helps minimize the danger of cross-classification and mis-classification of samples, and helps improve the flexibility with which parts of a corpus may be characterized for different applications.

Anthologies and collections are often treated as texts in their own right, if only for historical reasons. In conventional publishing, at least, anthologies are published as units, with single editorial responsibility and common front and back matter which may need to be included in their electronic encodings. The texts collected in the anthology, of course, may also need to be identifiable as distinct individual objects for study.

Poem cycles, epistolary novels, and epistolary essays differ from anthologies in that they are often written as single works, by single authors, for single occasions; nevertheless, it can be useful to treat their constituent parts as individual texts, as well as the cycle itself. Structurally, therefore, they may be treated in the same way as anthologies: in both cases, the body of the text is composed largely of other texts.

The <group> element is provided to simplify the encoding of collections, anthologies, and cyclic works; as noted above, the <group> element can also be used to record the potentially complex internal structure of language corpora. For a full description, see chapter *4. Default Text Structure*.

Some composite texts, finally, are neither corpora, nor anthologies, nor cyclic works: they are otherwise unitary texts within which other texts are embedded. In general, they may be treated in the same way as unitary texts, using the normal <TEI> and <body> elements. The embedded text itself may be encoded using the <text> element, which may occur within quotations or between paragraphs or other chunk-level elements inside the sections of a larger text. For further discussion, see chapter *4. Default Text Structure*.

All composite texts share the characteristic that their different component texts may be of structurally similar or dissimilar types. If all component texts may all be encoded using the same module, then no problem arises. If however they require different modules, then these must be included in the schema. This process is described in more detail in section *1.1. TEI Modules*.

15.2 Contextual Information

Contextual information is of particular importance for collections or corpora composed of samples from a variety of different kinds of text. Examples of such contextual information include: the age, sex, and geographical origins of participants in a language interaction, or their socio-economic status; the cost and publication data of a newspaper; the topic, register or factuality of an extract from a textbook. Such information may be of the first importance, whether as an organizing principle in creating a corpus (for example, to ensure that the range of values in such a parameter is evenly represented throughout the corpus, or represented proportionately to the population being sampled), or as a selection criterion in analysing the corpus (for example, to investigate the language usage of some particular vector of social characteristics).

Such contextual information is potentially of equal importance for unitary texts, and these Guidelines accordingly make no particular distinction between the kinds of information which should be gathered for unitary and for composite texts. In either case, the information should be recorded in the appropriate section of a TEI Header, as described in chapter *2. The TEI Header*. In the case of language corpora, such information may be gathered together in the overall corpus header, or split across all the component texts of a corpus, in their individual headers, or divided between the two. The association between an individual corpus text and the contextual information applicable to it may be made in a number of ways, as further discussed in section *15.3. Associating Contextual Information with a Text* below.

Chapter *2. The TEI Header*, which should be read in conjunction with the present section, describes in full the range of elements available for the encoding of information relating to the electronic file itself, for example its bibliographic description and those of the source or sources from which it was derived (see section *2.2. The File Description*); information about the encoding practices followed with the corpus, for example its design principles, editorial practices, reference system, etc. (see section *2.3. The Encoding Description*); more detailed descriptive information about the creation and content of the corpus, such as the languages used within it and any descriptive classification system used (see section *2.4. The Profile Description*); and version information documenting any changes made in the electronic text (see section *2.5. The Revision Description*).

In addition to the elements defined by chapter *2. The TEI Header*, several other elements can be used in the TEI header if the additional module defined by this chapter is invoked. These additional tags make it possible to characterize the social or other situation within which a language interaction takes place or is experienced, the physical setting of a language interaction, and the participants in it. Though this information may be relevant to, and provided for, unitary texts as well as for collections or corpora, it is more often recorded for

the components of systematically developed corpora than for isolated texts, and thus this module is referred to as being 'for language corpora'.

When the module defined in this chapter is included in a schema, a number of additional elements become available within the <profileDesc> element of the TEI Header (discussed in section *2.4. The Profile Description*).

<textDesc> (text description) provides a description of a text in terms of its situational parameters.

<particDesc> (participation description) describes the identifiable speakers, voices, or other participants in a linguistic interaction.

<settingDesc> (setting description) describes the setting or settings within which a language interaction takes place, either as a prose description or as a series of setting elements.

These elements, members of the model.profileDescPart, are discussed in the remainder of the chapter.

15.2.1 The Text Description

The <textDesc> element provides a full description of the situation within which a text was produced or experienced, and thus characterizes it in a way relatively independent of any *a priori* theory of text-types. It is provided as an alternative or a supplement to the common use of descriptive taxonomies used to categorize texts, which is fully described in section *2.4.3. The Text Classification*, and section *2.3.6. The Classification Declaration*. The description is organized as a set of values and optional prose descriptions for the following eight situational parameters, each represented by one of the following eight elements:

<channel> (primary channel) describes the medium or channel by which a text is delivered or experienced. For a written text, this might be print, manuscript, e-mail, etc.; for a spoken one, radio, telephone, face-to-face, etc.

@**mode** specifies the mode of this channel with respect to speech and writing.

<constitution> describes the internal composition of a text or text sample, for example as fragmentary, complete, etc.

@**type** specifies how the text was constituted.

<derivation> describes the nature and extent of originality of this text.

@**type** categorizes the derivation of the text.

<domain> (domain of use) describes the most important social context in which the text was realized or for which it is intended, for example private vs. public, education, religion, etc.

@**type** categorizes the domain of use.

<factuality> describes the extent to which the text may be regarded as imaginative or non-imaginative, that is, as describing a fictional or a non-fictional world.

@**type** categorizes the factuality of the text.

<interaction> describes the extent, cardinality and nature of any interaction among those producing and experiencing the text, for example in the form of response or interjection, commentary, etc.

@**type** specifies the degree of interaction between active and passive participants in the text.

@**active** specifies the number of active participants (or addressors) producing parts of the text.

@**passive** specifies the number of passive participants (or addressees) to whom a text is directed or in whose presence it is created or performed.

<preparedness> describes the extent to which a text may be regarded as prepared or spontaneous.

@**type** a keyword characterizing the type of preparedness.

<purpose> characterizes a single purpose or communicative function of the text.

@**type** specifies a particular kind of purpose.

@**degree** specifies the extent to which this purpose predominates.

These elements constitute a model class called model.textDescPart; new parameters may be defined by defining new elements and adding them to that class, as further described in *23.2. Personalization and Customization*.

By default, a text description will contain each of the above elements, supplied in the order specified. Except for the <purpose> element, which may be repeated to indicate multiple purposes, no element should appear more than once within a single text description. Each element may be empty, or may contain a brief qualification or more detailed description of the value expressed by its attributes. It should be noted that some texts, in particular literary ones, may resist unambiguous classification in some of these dimensions; in such cases, the situational parameter in question should be given the content 'not applicable' or an equivalent phrase.

Texts may be described along many dimensions, according to many different taxonomies. No generally accepted consensus as to how such taxonomies should be defined has yet emerged, despite the best efforts of many corpus linguists, text linguists, sociolinguists, rhetoricians, and literary theorists over the years. Rather than attempting the task of proposing a single taxonomy of text-types (or the equally impossible one of enumerating all those which have been proposed previously), the closed set of situational parameters described above can be used in combination to supply useful distinguishing descriptive features of individual texts, without insisting on a system of discrete high-level text-types. Such text-types may however be used in combination with the parameters proposed here, with the advantage that the internal structure of each such text-type can be specified in terms of the parameters proposed. This approach has the following analytical advantages:[1]

- it enables a relatively continuous characterization of texts (in contrast to discrete categories based on type or topic)
- it enables meaningful comparisons across corpora
- it allows analysts to build and compare their own text-types based on the particular parameters of interest to them
- it is equally applicable to spoken, written, or signed texts

Two alternative approaches to the use of these parameters are supported by these Guidelines. One is to use pre-existing taxonomies such as those used in subject classification or other types of text categorization. Such taxonomies may also be appropriate for the description of the topics addressed by particular texts. Elements for this purpose are described in section *2.4.3. The Text Classification*, and elements for defining or declaring such classification schemes in section *2.3.6. The Classification Declaration*. A second approach is to develop an application-specific set of feature structures and an associated feature system declaration, as described in chapters *18. Feature Structures* and *18.11. Feature System Declaration*.

Where the organizing principles of a corpus or collection so permit, it may be convenient to regard a particular set of values for the situational parameters listed in this section as forming a text-type in its own right; this may also be useful where the same set of values applies to several texts within a corpus. In such a case, the set of text-types so defined should be regarded as a taxonomy. The mechanisms described in section *2.3.6. The Classification Declaration* may be used to define hierarchic taxonomies of such text-types, provided that the <catDesc> component of the <category> element contains a <textDesc> element rather than a prose description. Particular texts may then be associated with such definitions using the mechanisms described in sections *2.4.3. The Text Classification*.

Using these situational parameters, an informal domestic conversation might be characterized as follows:

[1] Schemes similar to that proposed here were developed in the 1960s and 1970s by researchers such as Hymes, Halliday, and Crystal and Davy, but have rarely been implemented; one notable exception being the pioneering work on the Helsinki Diachronic Corpus of English, on which see Kytö and Rissanen (1988)

```
<textDesc n="Informal domestic conversation">
  <channel mode="s">informal face-to-face conversation</channel>
  <constitution type="single">each text represents a continuously
     recorded interaction among the specified participants
  </constitution>
  <derivation type="original"/>
  <domain type="domestic">plans for coming week, local affairs</domain>
  <factuality type="mixed">mostly factual, some jokes</factuality>
  <interaction type="complete" active="plural" passive="many"/>
  <preparedness type="spontaneous"/>
  <purpose type="entertain" degree="high"/>
  <purpose type="inform" degree="medium"/>
</textDesc>
```

The following example demonstrates how the same situational parameters might be used to characterize a novel:

```
<textDesc n="novel">
  <channel mode="w">print; part issues</channel>
  <constitution type="single"/>
  <derivation type="original"/>
  <domain type="art"/>
  <factuality type="fiction"/>
  <interaction type="none"/>
  <preparedness type="prepared"/>
  <purpose type="entertain" degree="high"/>
  <purpose type="inform" degree="medium"/>
</textDesc>
```

15.2.2 The Participant Description

The <particDesc> element in the <profileDesc> element provides additional information about the participants in a spoken text or, where this is judged appropriate, the persons named or depicted in a written text. When the detailed elements provided by the namesdates module described in *13. Names, Dates, People, and Places* are included in a schema, this element can contain detailed demographic or descriptive information about individual speakers or groups of speakers, such as their names or other personal characteristics. Individually identified persons may also identified by a code which can then be used elsewhere within the encoded text, for example as the value of a **who** attribute.

It should be noted that although the terms speaker or participant are used throughout this section, it is intended that the same mechanisms may be used to characterize fictional personæ or 'voices' within a written text, except where otherwise stated. For the purposes of analysis of language usage, the information specified here should be equally applicable to written, spoken, or signed texts.

The element <particDesc> contains a description of the participants in an interaction, which may be supplied as straightforward prose, possibly containing a list of names, encoded using the usual <list> and <name> elements, or alternatively using the more specific and detailed <listPerson> element provided by the namesdates module described in *13. Names, Dates, People, and Places*.

For example, a participant in a recorded conversation might be described informally as follows:

```
<particDesc xml:id="p2">
  <p>Female informant, well-educated, born in Shropshire UK, 12 Jan
```

```
    1950, of unknown occupation. Speaks French fluently.
    Socio-Economic status B2 in the PEP classification scheme.</p>
</particDesc>
```

Alternatively, when the namesdates module is included in a schema, information about the same participant described above might be provided in a more structured way as follows:

```
<person sex="2" age="mid">
  <birth when="1950-01-12">
    <date>12 Jan 1950</date>
    <name type="place">Shropshire, UK</name>
  </birth>
  <langKnowledge tags="en fr">
    <langKnown level="first" tag="en">English</langKnown>
    <langKnown tag="fr">French</langKnown>
  </langKnowledge>
  <residence>Long term resident of Hull</residence>
  <education>University postgraduate</education>
  <occupation>Unknown</occupation>
  <socecStatus scheme="#pep" code="#b2"/>
</person>
```

An identified character in a drama or a novel may also be regarded as a participant in this sense, and encoding using the same techniques:[2]

```
<particDesc>
  <p>The chief speaking characters in this novel are
  <list>
     <item xml:id="EMWOO">
       <name>Emma Woodhouse</name>
     </item>
     <item xml:id="DARCY">
       <name>Mr Darcy</name>
     </item>
<!-- ... -->
   </list>
  </p>
</particDesc>
```

Here, the characters are simply listed without the detailed structure which use of the <listPerson> element permits.

15.2.3 The Setting Description

The <settingDesc> element is used to describe the setting or settings in which language interaction takes place. It may contain a prose description, analogous to a stage description at the start of a play, stating in broad terms the locale, or a more detailed description of a series of such settings.

Each distinct setting is described by means of a <setting> element.

<setting> describes one particular setting in which a language interaction takes place.

[2] It is particularly useful to define participants in a dramatic text in this way, since it enables the **who** attribute to be used to link <sp> elements to definitions for their speakers; see further section *7.2.2. Speeches and Speakers*.

Individual settings may be associated with particular participants by means of the optional **who** attribute which this element inherits as a member of the att.ascribed if, for example, participants are in different places. This attribute identifies one or more individual participants or participant groups, as discussed earlier in section *15.2.2. The Participant Description*. If this attribute is not specified, the setting details provided are assumed to apply to all participants represented in the language interaction. Note however that it is not possible to encode different settings for the same participant: a participant is deemed to be a person within a specific setting.

The <setting> element may contain either a prose description or a selection of elements from the classes model.nameLike.agent, model.dateLike, or model.settingPart. By default, when the module definded by this chapter is included in a schema, these classes thus provide the following elements :

<name> (name, proper noun) contains a proper noun or noun phrase.
  @**type** indicates the type of the object which is being named by the phrase.

<date> contains a date in any format.

<time> contains a phrase defining a time of day in any format.

<locale> contains a brief informal description of the kind of place concerned, for example: a room, a restaurant, a park bench, etc.

<activity> contains a brief informal description of what a participant in a language interaction is doing other than speaking, if anything.

Additional more specific naming elements such as <orgName> or <persName> may also be available if the namesdates module is also included in the schema.

The following example demonstrates the kind of background information often required to support transcriptions of language interactions, first encoded as a simple prose narrative:

```
<settingDesc>
  <p>The time is early spring, 1989. P1 and P2 are playing on the rug
     of a suburban home in Bedford. P3 is doing the washing up at the
     sink. P4 (a radio announcer) is in a broadcasting studio in
     London.</p>
</settingDesc>
```

The same information might be represented more formally in the following way:

```
<settingDesc>
  <setting  who="#p1 #p2">
    <name  type="city">Bedford</name>
    <name  type="region">UK: South East</name>
    <date>early spring, 1989</date>
    <locale>rug of a suburban home</locale>
    <activity>playing</activity>
  </setting>
  <setting  who="#p3">
    <name  type="city">Bedford</name>
    <name  type="region">UK: South East</name>
    <date>early spring, 1989</date>
    <locale>at the sink</locale>
    <activity>washing-up</activity>
  </setting>
  <setting  who="#p4">
    <name  type="place">London, UK</name>
    <time>unknown</time>
    <locale>broadcasting studio</locale>
```

```
    <activity>radio performance</activity>
  </setting>
</settingDesc>
```

Again, a more detailed encoding for places is feasible if the namesdates module is included in the schema. The above examples assume that only the general purpose <name> element supplied in the core module is available.

15.3 Associating Contextual Information with a Text

This section discusses the association of the contextual information held in the header with the individual elements making up a TEI text or corpus. Contextual information is held in elements of various kinds within the TEI header, as discussed elsewhere in this section and in chapter *2. The TEI Header*. Here we consider what happens when different parts of a document need to be associated with different contextual information of the same type, for example when one part of a document uses a different encoding practice from another, or where one part relates to a different setting from another. In such situations, there will be more than one instance of a header element of the relevant type.

The TEI scheme allow for the following possibilities:

- A given element may appear in the corpus header only, in the header of one or more texts only, or in both places
- There may be multiple occurrences of certain elements in either corpus or text header.

To simplify the exposition, we deal with these two possibilities separately in what follows; however, they may be combined as desired.

15.3.1 Combining Corpus and Text Headers

A TEI-conformant document may have more than one header only in the case of a TEI corpus, which must have a header in its own right, as well as the obligatory header for each text. Every element specified in a corpus-header is understood as if it appeared within every text header in the corpus. An element specified in a text header but not in the corpus header supplements the specification for that text alone. If any element is specified in both corpus and text headers, the corpus header element is over-ridden for that text alone.

The <titleStmt> for a corpus text is understood to be prefixed by the <titleStmt> given in the corpus header. All other optional elements of the <fileDesc> should be omitted from an individual corpus text header unless they differ from those specified in the corpus header. All other header elements behave identically, in the manner documented below. This facility makes it possible to state once for all in the corpus header each piece of contextual information which is common to the whole of the corpus, while still allowing for individual texts to vary from this common denominator.

For example, the following schematic shows the structure of a corpus comprising three texts, the first and last of which share the same encoding declaration. The second one has its own encoding declaration

```
<teiCorpus>
  <teiHeader>
    <fileDesc>
<!-- corpus file description-->
    </fileDesc>
    <encodingDesc>
<!-- default encoding description -->
    </encodingDesc>
    <revisionDesc>
```

```
<!-- corpus revision description -->
   </revisionDesc>
  </teiHeader>
  <TEI>
    <teiHeader>
     <fileDesc>
<!-- file description for this corpus text -->
     </fileDesc>
    </teiHeader>
    <text>
<!-- first corpus text -->
    </text>
  </TEI>
  <TEI>
    <teiHeader>
     <fileDesc>
<!-- file description for this corpus text -->
     </fileDesc>
     <encodingDesc>
<!-- encoding description for this corpus text, over-riding the default -->
     </encodingDesc>
    </teiHeader>
    <text>
<!-- second corpus text -->
    </text>
  </TEI>
  <TEI>
    <teiHeader>
     <fileDesc>
<!-- file description for third corpus text -->
     </fileDesc>
    </teiHeader>
    <text>
<!-- third corpus text -->
    </text>
  </TEI>
</teiCorpus>
```

15.3.2 Declarable Elements

Certain of the elements which can appear within a TEI Header are known as declarable elements. These elements have in common the fact that they may be linked explicitly with a particular part of a text or corpus by means of a **decls** attribute on that element. This linkage is used to over-ride the default association between declarations in the header and a corpus or corpus text. The only header elements which may be associated in this way are those which would not otherwise be meaningfully repeatable.

Declarable elements are all members of the class att.declarable; the corresponding declaring elements are all members of the the class att.declarable

att.declarable provides attributes for those elements in the TEI Header which may be independently selected by means of the special purpose **decls** attribute.

@**default** indicates whether or not this element is selected by default when its parent is selected.

att.declaring provides attributes for elements which may be independently associated with a particular declarable element within the header, thus overriding the inherited default for that element.

@**decls** identifies one or more declarable elements within the header, which are understood to apply to the element bearing this attribute and its content.

An alphabetically ordered list of declarable elements follows:

<availability> supplies information about the availability of a text, for example any restrictions on its use or distribution, its copyright status, etc.

<bibl> (bibliographic citation) contains a loosely-structured bibliographic citation of which the sub-components may or may not be explicitly tagged.

<biblFull> (fully-structured bibliographic citation) contains a fully-structured bibliographic citation, in which all components of the TEI file description are present.

<biblStruct> (structured bibliographic citation) contains a structured bibliographic citation, in which only bibliographic sub-elements appear and in a specified order.

<broadcast> describes a broadcast used as the source of a spoken text.

<correction> (correction principles) states how and under what circumstances corrections have been made in the text.

<editorialDecl> (editorial practice declaration) provides details of editorial principles and practices applied during the encoding of a text.

<equipment> provides technical details of the equipment and media used for an audio or video recording used as the source for a spoken text.

<graphic/> indicates the location of an inline graphic, illustration, or figure.

<hyphenation> summarizes the way in which hyphenation in a source text has been treated in an encoded version of it.

<interpretation> describes the scope of any analytic or interpretive information added to the text in addition to the transcription.

<langUsage> (language usage) describes the languages, sublanguages, registers, dialects, etc. represented within a text.

<listBibl> (citation list) contains a list of bibliographic citations of any kind.

<normalization> indicates the extent of normalization or regularization of the original source carried out in converting it to electronic form.

<particDesc> (participation description) describes the identifiable speakers, voices, or other participants in a linguistic interaction.

<projectDesc> (project description) describes in detail the aim or purpose for which an electronic file was encoded, together with any other relevant information concerning the process by which it was assembled or collected.

<quotation> specifies editorial practice adopted with respect to quotation marks in the original.

<recording> (recording event) details of an audio or video recording event used as the source of a spoken text, either directly or from a public broadcast.

<samplingDecl> (sampling declaration) contains a prose description of the rationale and methods used in sampling texts in the creation of a corpus or collection.

<scriptStmt> (script statement) contains a citation giving details of the script used for a spoken text.

<segmentation> describes the principles according to which the text has been segmented, for example into sentences, tone-units, graphemic strata, etc.

<sourceDesc> (source description) supplies a description of the source text(s) from which an electronic text was derived or generated.

<stdVals> (standard values) specifies the format used when standardized date or number values are supplied.

<textClass> (text classification) groups information which describes the nature or topic of a text in terms of a standard classification scheme, thesaurus, etc.

<textDesc> (text description) provides a description of a text in terms of its situational parameters.

All of the above elements may be multiply defined within a single header, that is, there may be more than one instance of any declarable element type at a given level. When this occurs, the following rules apply:

- every declarable element must bear a unique identifier
- for each different type of declarable element which occurs more than once within the same parent element, exactly one element must be specified as the default, by means of the **default** attribute

In the following example, an editorial declaration contains two possible <correction> policies, one identified as **CorPol1** and the other as **CorPol2**. Since there are two, one of them (in this case **CorPol1**) must be specified as the default:

```
<editorialDecl>
  <correction xml:id="CorPol1" default="true">
    <p> ... </p>
  </correction>
  <correction xml:id="CorPol2">
    <p> ... </p>
  </correction>
  <normalization xml:id="n1">
    <p> ... </p>
    <p> ... </p>
  </normalization>
</editorialDecl>
```

For texts associated with the header in which this declaration appears, correction method **CorPol1** will be assumed, unless they explicitly state otherwise. Here is the structure for a text which does state otherwise:

```
<text>
  <body>
    <div1 n="d1"/>
    <div1 n="d2" decls="#CorPol2"/>
    <div1 n="d3"/>
  </body>
</text>
```

In this case, the contents of the divisions D1 and D3 will both use correction policy **CorPol1**, and those of division D2 will use correction policy **CorPol2**.

The **decls** attribute is defined for any element which is a member of the class declaring. This includes the major structural elements <text>, <group>, and , as well as smaller structural units, down to the level of paragraphs in prose, individual utterances in spoken texts, and entries in dictionaries. However, TEI recommended practice is to limit the number of multiple declarable elements used by a document as far as possible, for simplicity and ease of processing.

The identifier or identifiers specified by the **decls** attribute are subject to two further restrictions:

- An identifier specifying an element which contains multiple instances of one or more other elements should be interpreted as if it explicitly identified the elements identified as the default in each such set of repeated elements

- Each element specified, explicitly or implicitly, by the list of identifiers must be of a different kind.

To demonstrate how these rules operate, we now expand our earlier example slightly:

```
<encodingDesc>
  <editorialDecl xml:id="ED1" default="true">
    <correction xml:id="C1A" default="true">
      <p> ... </p>
    </correction>
    <correction xml:id="C1B">
      <p> ... </p>
    </correction>
    <normalization xml:id="N1">
      <p> ... </p>
      <p> ... </p>
    </normalization>
  </editorialDecl>
  <editorialDecl xml:id="ED2">
    <correction xml:id="C2A" default="true">
      <p> ... </p>
    </correction>
    <correction xml:id="C2B">
      <p> ... </p>
    </correction>
    <normalization xml:id="N2A">
      <p> ... </p>
    </normalization>
    <normalization xml:id="N2B" default="true">
      <p> ... </p>
    </normalization>
  </editorialDecl>
</encodingDesc>
```

This encoding description now has two editorial declarations, identified as **ED1** (the default) and **ED2**. For texts not specifying otherwise, **ED1** will apply. If **ED1** applies, correction method C1A and normalization method N1 apply, since these are the specified defaults within **ED1**. In the same way, for a text specifying **decls** as '**ED2**', correction C2A, and normalization N2B will apply.

A finer grained approach is also possible. A text might specify **<text decls='C2B N2A'>**, to 'mix and match' declarations as required. A tag such as **<text decls='ED1 ED2'>** would (obviously) be illegal, since it includes two elements of the same type; a tag such as **<text decls='ED2 C1A'>** is also illegal, since in this context **ED2** is synonymous with the defaults for that editorial declaration, namely **C2A N2B**, resulting in a list that identifies two correction elements (C1A and C2A).

15.3.3 Summary

The rules determing which of the declarable elements are applicable at any point may be summarized as follows:

1. If there is a single occurrence of a given declarable element in a corpus header, then it applies by default to all elements within the corpus.
2. If there is a single occurrence of a given declarable element in the text header, then it applies by default to all elements of that text irrespective of the contents of the corpus header.
3. Where there are multiple occurrences of declarable elements within either corpus or text header,
   - each must have a unique value specified as the value of its **xml:id** attribute;

- one only must bear a **default** attribute with the value **YES**.

4. It is a semantic error for an element to be associated with more than one occurrence of any declarable element.

5. Selecting an element which contains multiple occurrences of a given declarable element is semantically equivalent to selecting only those contained elements which are specified as defaults.

6. An association made by one element applies by default to all of its descendants.

15.4 Linguistic Annotation of Corpora

Language corpora often include analytic encodings or annotations, designed to support a variety of different views of language. The present Guidelines do not advocate any particular approach to linguistic annotation (or 'tagging'); instead a number of general analytic facilities are provided which support the representation of most forms of annotation in a standard and self-documenting manner. Analytic annotation is of importance in many fields, not only in corpus linguistics, and is therefore discussed in general terms elsewhere in the Guidelines.[3] The present section presents informally some particular applications of these general mechanisms to the specific practice of corpus linguistics.

15.4.1 Levels of Analysis

By linguistic annotation we mean here any annotation determined by an analysis of linguistic features of the text, excluding as borderline cases both the formal structural properties of the text (e.g. its division into chapters or paragraphs) and descriptive information about its context (the circumstances of its production, its genre, or medium). The structural properties of any TEI-conformant text should be represented using the structural elements discussed elsewhere in these Guidelines, for example in chapters *3. Elements Available in All TEI Documents* and *4. Default Text Structure*. The contextual properties of a TEI text are fully documented in the TEI Header, which is discussed in chapter *2. The TEI Header*, and in section *15.2. Contextual Information* of the present chapter.

Other forms of linguistic annotation may be applied at a number of levels in a text. A code (such as a word-class or part-of-speech code) may be associated with each word or token, or with groups of such tokens, which may be continuous, discontinuous, or nested. A code may also be associated with relationships (such as cohesion) perceived as existing between distinct parts of a text. The codes themselves may stand for discrete non-decomposable categories, or they may represent highly articulated bundles of textual features. Their function may be to place the annotated part of the text somewhere within a narrowly linguistic or discoursal domain of analysis, or within a more general semantic field, or any combination drawn from these and other domains.

The manner by which such annotations are generated and attached to the text may be entirely automatic, entirely manual, or a mixture. The ease and accuracy with which analysis may be automated may vary with the level at which the annotation is attached. The method employed should be documented in the <interpretation> element within the encoding description of the TEI Header, as described in section *2.3.3. The Editorial Practices Declaration*. Where different parts of a corpus have used different annotation methods, the **decls** attribute may be used to indicate the fact, as further discussed in section *15.3. Associating Contextual Information with a Text*.

An extended example of one form of linguistic analysis commonly practised in corpus linguistics is given in section *17.4. Linguistic Annotation*.

[3] See in particular chapters *16. Linking, Segmentation, and Alignment*, *17. Simple Analytic Mechanisms*, and *18. Feature Structures*.

15.5 Recommendations for the Encoding of Large Corpora

These Guidelines include proposals for the identification and encoding of a far greater variety of textual features and characteristics than is likely to be either feasible or desirable in any one language corpus, however large and ambitious. The reasoning behind this catholic approach is further discussed in chapter *iv About These Guidelines*. For most large-scale corpus projects, it will therefore be necessary to determine a subset of TEI recommended elements appropriate to the anticipated needs of the project, as further discussed in chapter *23.2. Personalization and Customization*; these mechanisms include the ability to exclude selected element types, add new element types, and change the names of existing elements. A discussion of the implications of such changes for TEI conformance is provided in chapter *23.3. Conformance*.

Because of the high cost of identifying and encoding many textual features, and the difficulty in ensuring consistent practice across very large corpora, encoders may find it convenient to divide the set of elements to be encoded into the following three categories:

required texts included within the corpus will always encode textual features in this category, should they exist in the text

recommended textual features in this category will be encoded wherever economically and practically feasible; where present but not encoded, a note in the header should be made.

optional textual features in this category may or may not be encoded; no conclusion about the absence of such features can be inferred from the absence of the corresponding element in a given text.

proscribed textual features in this category are deliberately not encoded; they may be transcribed as unmarked up text, or represented as <gap> elements, or silently omitted, as appropriate.

15.6 Module for Language Corpora

The module described in this chapter makes available the following components:

Module corpus: Corpus texts

- Elements defined: activity channel constitution derivation domain factuality interaction locale particDesc preparedness purpose setting settingDesc textDesc

The selection and combination of modules to form a TEI schema is described in *1.2. Defining a TEI Schema*.

Chapter 16

Linking, Segmentation, and Alignment

This chapter discusses a number of ways in which encoders may represent analyses of the structure of a text which are not necessarily linear or hierarchic. The module defined by this chapter provides for the following common requirements:

- to link disparate elements using the **xml:id** attribute (section *16.1. Links*);
- to link disparate elements without using the **xml:id** attribute (sections *16.2.1. Pointing Elsewhere*, *16.2.3. W3C element() Scheme*, and *16.2.4. TEI XPointer Schemes*);
- to segment text into elements convenient for the encoder and to mark arbitrary points within documents (section *16.3. Blocks, Segments, and Anchors*);
- to represent correspondence or alignment among groups of text elements, both those with content and those which are empty (section *16.4. Correspondence and Alignment*);[1]
- to synchronize elements of a text, that is to represent temporal correspondences and alignments among text elements (section *16.5. Synchronization*) and also to align them with specific points in time (section *16.5.2. Placing Synchronous Events in Time*);
- to specify that one text element is identical to or a copy of another (section *16.6. Identical Elements and Virtual Copies*);
- to aggregate possibly noncontinguous elements (section *16.7. Aggregation*);
- to specify that different elements are alternatives to one another and to express preferences among the alternatives (section *16.8. Alternation*);
- to store markup separately from the data it describes (section *16.9. Stand-off Markup*);
- to associate segments of a text with interpretations or analyses of their significance (section *16.10. Connecting Analytic and Textual Markup*).

These facilities all use the same set of techniques based on the W3C XPointer framework (Grosso et al. (eds.) (2003)) This provides a variety of schemes; the most convenient of which, and that recommended by these Guidelines, makes use of the global **xml:id** attribute, as defined in section *1.3.1.1. Global Attributes*, and introduced in the section of *v A Gentle Introduction to XML* titled *v.5.2 Identifiers and indicators* . When the linking module is included in a schema, the attribute class att.global is extended to include eight additional attributes to support the various kinds of linking listed above. Each of these attributes is introduced in the appropriate section below. In addition, for many of the topics discussed, a choice of methods of encoding is

[1] We use the term alignment as a special case for the more general notion of correspondence. Using A as a short form for 'an element with its attribute **xml:id** set to the value **A**', and suppose elements A1, A2, and A3 occur in that order and form one group, while elements B1, B2, and B3 occur in that order and form another group. Then a relation in which A1 corresponds to B1, A2 corresponds to B2, and A3 corresponds to B3 is an alignment. On the other hand, a relation in which A1 corresponds to B2, B1 to C2, and C1 to A2 is not an alignment.

offered, ranging from simple but less general ones, which use attribute values only, to more elaborate and more general ones, which use specialized elements.

16.1 Links

We say that one element points to others if the first has an attribute whose value is a reference to the others: such an element is called a pointer element, or simply a pointer. Among the pointers that have been introduced up to this point in these Guidelines are <note>, <ref>, and <ptr>. These elements all indicate an association between one place in the document (the location of the pointer itself) and one or more others (the elements whose identifiers are specified by the pointer's **target** attribute). The module described in this chapter introduces a variation on this basic kind of pointer, known as a link, which specifies both 'ends' of an association. In addition, we define a syntax for representing locations in a document by a variety of means not dependent on the use of **xml:id** attributes.

16.1.1 Pointers and Links

In section *3.6. Simple Links and Cross-References* we introduced the simplest pointer elements, <ptr> and <ref>. Here we introduce additionally the <link> element, which represents an association between two (or more) locations by specifying each location explicitly. Its own location is irrelevant to the intended linkage.

<ptr/> (pointer) defines a pointer to another location.
- @**target** specifies the destination of the pointer by supplying one or more URI References

<ref> (reference) defines a reference to another location, possibly modified by additional text or comment.
- @**target** specifies the destination of the reference by supplying one or more URI References

<link/> defines an association or hypertextual link among elements or passages, of some type not more precisely specifiable by other elements.
- @**targets** specifies the identifiers of the elements or passages to be linked or associated.

The <ptr> element may be called a 'pure pointer', because its primary function is simply to point. A pointer sets up a connection between an element (which, in the case of a pure pointer, is simply a location in a document), and one or more others, known collectively as its target. The <ptr> and <ref> elements bear a **target** attribute (in the singular), because they point, conceptually, at a single target, even if that target may be discontinuous in the document. The <link> element bears a **targets** attribute (in the plural), because it specifies at least two targets, each of which is a unitary object. It may be thought of as a representing a double link between the objects specified.

As members of the class att.pointing, these elements share a common set of attributes:

att.pointing defines a set of attributes used by all elements which point to other elements by means of one or more URI references.
- @**type** categorizes the pointer in some respect, using any convenient set of categories.
- @**evaluate** specifies the intended meaning when the target of a pointer is itself a pointer.

Double connection among elements could also be expressed by a combination of pointer elements, for example, two <ptr> elements, or one <ptr> element and one <note> element. All that is required is that the value of the **target** (or other pointing) attribute of the one be the value of the **xml:id** attribute of the other. What the <link> element accomplishes is the handling of double connection by means of a single element. Thus, in the following encoding:

```
<ptr xml:id="sa-p1" target="#sa-p2"/>
<ptr xml:id="sa-p2" target="#sa-p1"/>
```

sa-p1 points to **sa-p2**, and **sa-p2** points to **sa-p1**. This is logically equivalent to the more compact encoding:

```
<link  targets="#sa-p1 #sa-p2"/>
```

As noted elsewhere, both **target** and **targets** attributes take as value one or more URI reference. In the simplest case, a URI reference might indicate an element in the current document (or in some other document) by supplying the value used for its global **xml:id** attribute. Pointing or linking to external documents and pointing and linking where identifiers are not available are implemented by more complex forms of URI references, as described below in section *16.2. Pointing Mechanisms*.

16.1.2 Using Pointers and Links

As an example of the use of these mechanisms which establish connections among elements, consider the practice (common in 18th century English verse and elsewhere) of providing footnotes citing parallel passages from classical authors. Such footnotes can of course simply be encoded using the <note> element (see section *3.8. Notes, Annotation, and Indexing*) without a **target** attribute, placed adjacent to the passage to which the note refers:[2]

```
<l>(Diff'rent our parties, but with equal grace</l>
<l>The Goddess smiles on Whig and Tory race,</l>
<l>
  <note  type="imitation"  place="foot"  anchored="false">
    <bibl>Virg. Æn. 10.</bibl>
    <quote>
      <l>Tros Rutulusve fuat; nullo discrimine habebo.</l>
      <l>—— Rex Jupiter omnibus idem.</l>
    </quote>
  </note>'Tis the same rope at sev'ral ends they twist,
</l>
<l>To Dulness, Ridpath is as dear as Mist)</l>
```

Source: [160]

This use of the <note> element can be called implicit pointing (or implicit linking). It relies on the juxtaposition of the note to the text being commented on for the connection to be understood. If it is felt that the mere juxtaposition of the note to the text does not make it sufficiently clear exactly what text segment is being commented on (for example, is it the immediately preceding line, or the immediately preceding two lines, or what?), or if it is decided to place the note at some distance from the text, then the pointing or the linking must be made explicit. We now consider various methods for doing that.

Firstly, a <ptr> element might be placed at an appropriate point within the text to link it with the annotation:

```
<l>(Diff'rent our parties, but with equal grace</l>
<l>The Goddess smiles on Whig and Tory race,
<ptr  rend="unmarked"  target="#note3.284"/>
</l>
<l>'Tis the same rope at sev'ral ends they twist,</l>
<l>To Dulness, Ridpath is as dear as Mist)</l>
<note
   xml:id="note3.284"
```

[2] The **type** attribute on the note is used to classify the notes using the typology established in the Advertisement to the work: 'The Imitations of the Ancients are added, to gratify those who either never read, or may have forgotten them; together with some of the Parodies, and Allusions to the most excellent of the Moderns.' In the source text, the text of the poem shares the page with two sets of notes, one headed 'Remarks' and the other 'Imitations'.

Their full-fed Heroes, their pacific May'rs,
Their annual trophies, and their monthly wars.
Tho' long my Party built on me their hopes,
For writing pamphlets, and for burning Popes;
(Diff'rent our parties, but with equal grace
The Goddeſs ſmiles on Whig and Tory race,
'Tis the ſame rope at ſev'ral ends they twiſt,
To Dulneſs, Ridpath is as dear as Miſt.)
Yet lo! in me what authors have to brag on!
Reduc'd at laſt to hiſs in my own dragon.
Avert it, Heav'n! that thou or Cibber e'er
Should wag two ſerpent tails in Smithfield fair.
Like the vile ſtraw that's blown about the ſtreets
The needy Poet ſticks to all he meets,
Coach'd, carted, trod upon, now looſe, now faſt,
In the Dog's tail his progreſs ends at laſt.

REMARKS.

VERSE 280. Annual trophies, on the *Lord Mayor's Day*; and monthly wars, in the *Artillery Ground*.

VERSE 281. *Tho' long my Party.*] *Settle*, like moſt Party-writers, was very uncertain in his political principles. He was employ'd to hold the pen in the *Character* of a *Popiſh ſucceſſor*, but afterwards printed his *Narrative* on the contrary ſide. He had managed the Ceremony of a famous Pope-burning on *Nov.* 17, 1680: then became a Trooper of King *James*'s army at *Hounſlow-heath*: After the Revolution he kept a Booth at *Bartlemew-fair*, where in his Droll call'd St. *George for England*, he acted in his old age in a Dragon of green leather of his own invention. He was at laſt taken into the Charter-houſe, and there dyed, aged about 60 years.

V. 285. *To Dulneſs*, Ridpath *is as dear as* Miſt.] *George Ridpath*, author for ſeveral years of the *Flying-Poſt*, a Whig-paper; *Nathaniel Miſt*, publiſher of the Weekly Journal, a Tory-paper.

IMITATIONS.

VERSE 283-84. —— *With equal grace Our Goddeſs ſmiles on Whig and Tory race.*] *Tros Rutulusve fuat, nullo diſcrimine habebo.* —— *Rex Jupiter omnibus idem.* Virg. Æn. 10.

P

```
    type="imitation"
    place="foot"
    anchored="false">
  <bibl>Virg. Æn. 10.</bibl>
  <quote>
    <l>Tros Rutulusve fuat; nullo discrimine habebo.</l>
    <l>—— Rex Jupiter omnibus idem.</l>
  </quote>
</note>
```

Source: [160]

The <note> element has been given an arbitrary identifier (**note3.284**) to enable it to be specified as the target of the pointer element. Because there is nothing in the text to signal the existence of the annotation, the **rend** attribute has been given the value **unmarked**.

Secondly, the **target** attribute of the <note> element can be used to point at its associated text, provided that an **xml:id** attribute has been supplied for the associated text:

```
<l  xml:id="l3.283">(Diff'rent our parties, but with equal grace</l>
<l  xml:id="l3.284">The Goddess smiles on Whig and Tory race,</l>
<l  xml:id="l3.285">'Tis the same rope at sev'ral ends they twist,</l>
<l  xml:id="l3.286">To Dulness, Ridpath is as dear as Mist)</l>
<!-- ... -->
```

Source: [160]

Given this encoding of the text itself, we can now link the various notes to it. In this case, the note itself contains a pointer to the place in the text which it is annotating; this could be encoded using a <ref> element, which bears a **target** attribute of its own and contains a (slightly misquoted) extract from the text marked as a <quote> element:

```
<note
    type="imitation"
    place="foot"
    anchored="false"
    target="#l3.284">
  <ref  rend="sc"  target="#l3.284">Verse 283–84.
  <quote>
     <l>——. With equal grace</l>
     <l>Our Goddess smiles on Whig and Tory race.</l>
   </quote>
  </ref>
  <bibl>Virg. Æn. 10.</bibl>
  <quote>
    <l>Tros Rutulusve fuat; nullo discrimine habebo.</l>
    <l>—— Rex Jupiter omnibus idem. </l>
  </quote>
</note>
```

Source: [160]

Combining these two approaches gives us the following associations:

- a pointer within one line indicates the note
- the note indicates the line

- a pointer within the note indicates the line

Note that we do not have any way of pointing from the line itself to the note: the association is implied by containment of the pointer. We do not as yet have a true double link between text and note. To achieve that we will need to supply identifiers for the annotations as well as for the verse lines, and use a <link> element to associate the two. Note that the <ptr> element and the **target** attribute on the <note> may now be dispensed with:

```
<note
   xml:id="n3.284"
   type="imitation"
   place="foot"
   anchored="false">
  <ref  rend="sc"  target="#l3.284">Verse 283–84.
  <quote>
     <l>——. With equal grace</l>
     <l>Our Goddess smiles on Whig and Tory race.</l>
   </quote>
  </ref>
  <bibl>Virg. Æn. 10.</bibl>
  <quote>
    <l>Tros Rutulusve fuat; nullo discrimine habebo.</l>
    <l>—— Rex Jupiter omnibus idem. </l>
  </quote>
</note>
<link  targets="#n3.284 #l3.284"/>
```

Source: [160]

The **targets** attribute of the <link> element here bears the identifiers of the note followed by that of the verse line. For completeness, we could also allocate an identifier to the reference within the note and encode the association between it and the verse line in the same way:

```
<note
   xml:id="nt3.284"
   type="imitation"
   place="foot"
   anchored="false">
  <ref  rend="sc"  xml:id="r3.284"  target="#l3.284">Verse 283–84.
  <quote>
     <l>——. With equal grace</l>
     <l>Our Goddess smiles on Whig and Tory race.</l>
   </quote>
  </ref>
<!-- ... -->
</note>
<!-- ... -->
<link  targets="#r3.284 #l3.284"/>
```

Source: [160]

Indeed, the two <link>s could be combined into one, as follows:

```
<link  targets="#n3.284 #r3.284 #l3.284"/>
```

16.1.3 Groups of Links

Clearly, there are many reasons for which an encoder might wish to represent a link or association between different elements. For some of them, specific elements are provided in these Guidelines; some of these are discussed elsewhere in the present chapter. The <link> element is a general purpose element which may be used for any kind of association. The element <linkGrp> may be used to group links of a particular type together in a single part of the document; such a collection may be used to represent what is sometimes referred to in the literature of Hypertext as a web, a term introduced by the Brown University FRESS project in 1969.

<linkGrp> (link group) defines a collection of associations or hypertextual links.

As a member of the class att.pointing.group, this element shares the following attributes with other members of that class:

att.pointing.group defines a set of attributes common to all elements which enclose groups of pointer elements.

- @**domains** optionally specifies the identifiers of the elements within which all elements indicated by the contents of this element lie.
- @**targFunc** (target function) describes the function of each of the values of the **targets** attribute of the enclosed <link>, <join>, or <alt> tags.

It is also a member of the att.pointing class, and therefore also carries the attributes specified in section *16.1.1. Pointers and Links* above, in particular the **type** attribute:

att.pointing defines a set of attributes used by all elements which point to other elements by means of one or more URI references.

- @**type** categorizes the pointer in some respect, using any convenient set of categories.

The <linkGrp> element provides a convenient way of establishing a default for the **type** attribute on a group of links of the same type: by default, the **type** attribute on a <link> element has the same value as that given for **type** on the enclosing <linkGrp>.

Typical software might hide a web entirely from the user, but use it as a source of information about links, which are displayed independently at their referenced locations. Alternatively, software might provide a direct view of the link collection, along with added functions for manipulating the collection, as by filtering, sorting, and so on. To continue our previous example, this text contains many other notes of a kind similar to the one shown above. Here are a few more of the lines to which annotations have to be attached, followed by the annotations themselves:

```
<l xml:id="l2.79">A place there is, betwixt earth, air and seas</l>
<l xml:id="l2.80">Where from Ambrosia, Jove retires for ease.</l>
<!-- ... -->
<l xml:id="l2.88">Sign'd with that Ichor which from Gods distills.</l>
<!-- ... -->
<note xml:id="n2.79" place="foot" anchored="false">
  <bibl>Ovid Met. 12.</bibl>
  <quote xml:lang="la">
    <l>Orbe locus media est, inter terrasq; fretumq;</l>
    <l>Cœlestesq; plagas –</l>
  </quote>
</note>
<note xml:id="n2.88" place="foot" anchored="false"> Alludes to <bibl>Homer, Iliad 5</bibl> ...
</note>
```

To avoid having to repeat the specification of **type** as **imitation** on each <note>, we may specify it once for all on a <linkGrp> element containing all links of this type.

```
<linkGrp type="imitation">
  <link targets="#n2.79 #l2.79"/>
  <link targets="#n2.88 #l2.88"/>
  <link targets="#n3.284 #l3.284"/>
</linkGrp>
```

Additional information for applications that use <linkGrp> elements can be provided by means of special attributes. First, the **domains** attribute can be used to identify the text elements within which the individual targets of the links are to be found. Suppose that the text under discussion is organized into a <body> element, containing the text of the poem, and a <back> element containing the notes. Then the **domains** attribute can have as its value the identifiers of the <body> and the <back>, to enable an application to verify that the link targets are in fact contained by appropriate elements, or to limit its search space:

```
<!-- ... --><linkGrp type="imitation" domains="dunciad dunnotes">
  <link targets="#n2.79 #l2.79"/>
  <link targets="#n2.88 #l2.88"/>
<!-- ... -->
  <link targets="#n3.284 #l3.284"/>
<!-- ... -->
</linkGrp>
```

Note that there must be a single parent element for each 'domain'; if some notes are contained by a section with identifier **dunnotes**, and others by a section with identifier **dunimits**, an intermediate pointer must be provided (as described in section *16.1.4. Intermediate Pointers*) within the <linkGrp> and its identifier used instead.

Next, the **targFunc** attribute can be used to provide further information about the role or function of the various targets specified for each link in the group. The value of the **targFunc** attribute is a list of names (formally, name tokens), one for each of the targets in the link; these names can be chosen freely by the encoder, but their significance should be documented in the encoding declaration in the header.[3] In the current example, we might think of the note as containing the source of the imitation and the verse line as containing the goal of the imitation. Accordingly, we can specify the <linkGrp> in the preceding example thus:

```
<linkGrp type="imitation" domains="dunciad dunnotes" targFunc="source goal">
  <link targets="#n2.79 #l2.79"/>
  <link targets="#n2.88 #l2.88"/>
<!-- ... -->
  <link targets="#n3.284 #l3.284"/>
<!-- ... -->
</linkGrp>
```

16.1.4 Intermediate Pointers

In the preceding examples, we have shown various ways of linking an annotation and a single verse line. However, the example cited in fact requires us to encode an association between the note and a *pair* of verse lines (lines 284 and 285).

There are a number of possible ways of correcting this error: one could use the **target** and **targetEnd** attributes of the <note> element to delimit the span to which the note applies (see further section *3.8. Notes,*

[3] Since no special element is provided for this purpose in the present version of these Guidelines, the information should be supplied as a series of paragraphs at the end of the <encodingDesc> element described in section *2.3. The Encoding Description*.

Annotation, and Indexing). Alternatively one could create an element to encode the couplet itself and assign it an **xml:id** attribute, which can then be linked to the <note> and <ref> elements. This could be done either explicitly by means of an <lg> element, as defined in section *3.12.1. Core Tags for Verse* or implicitly, by means of the <join> element discussed in section *16.7. Aggregation*.

A third possibility however, is to use an 'intermediate pointer' as follows:

```
<ptr xml:id="l3.283-284" target="#l3.283 #l3.284"/>
```

When the **target** attribute of a <ptr> or <ref> element specifies more than one element, the indicated elements are intended to be combined or aggregated in some way to produce the object of the pointer. (Such aggregation is however the task of a processing application, and cannot be defined simply by the markup). The **xml:id** attribute provides an identifier which can then be linked to the <note> and <ref> elements:

```
<link evaluate="all" targets="#n3.284 #r3.284 #l3.283-284"/>
```

The **all** value of **evaluate** is used on the <link> element to specify that any pointer encountered as a target of that element is itself evaluated. If **evaluate** had the value **none**, the link target would be the pointer itself, rather than the objects it points to.

Where a <linkGrp> element is used to group a collection of <link> elements, any intermediate pointer elements used by those <link> elements should be included within the <linkGrp>.

16.2 Pointing Mechanisms

This section introduces more formally the pointing mechanisms available in the TEI. In addition to those discussed so far, the TEI provides methods of pointing:

- into documents other than the current document;
- to a particular element in a document other than the current document using its **xml:id**;
- to a particular element whether in the current document or not, using its position in the XML element tree;
- at arbitrary content in any XML document using TEI-defined XPointer schemes.

All TEI attributes used to point at something else are declared as having the datatype data.pointer, which is defined as a URI reference[4]; the cases so far discussed are all simple examples of a URI reference. Another familiar example is the mechanism used in XHTML to create represent hypertext links by means of the XHTML **href** attribute. A URI reference can reference the whole of an XML resource such as a document or an XML element, or a sub-portion of such a resource, identified by means of an appropriate fragment identifier. Technically speaking, the 'fragment identifier' is that portion of a URI reference following the first unescaped '#' character; in practice, it provides a means of accessing some part of the resource described by the URI which is less than the whole.

The first three of the following subsections provide only a brief overview and some examples of the W3C mechanisms recommended. More detailed information on the use of these mechanisms is readily available elsewhere.

16.2.1 Pointing Elsewhere

Like the ubiquitous if misnamed XHTML pointing attribute **href**, the TEI pointing attributes can point to a document that is not the current document (the one that contains the pointing element) whether it is in the

[4] The URI (Universal Resource Indicator) is defined in RFC 3986

same local filesystem as the current document, or on a different system entirely. In either case, the pointing can be accomplished absolutely (using the entire address of the target document) or relatively (using an address relative to the current base URI in force). The 'current base URI' is defined according to Marsh 2001. In general the current base URI in force is the value of the **xml:base** attribute of the closest ancestor that has one. If there is none, the base URI is that of the current document.

The following example demonstrates an absolute URI reference that points to a remote document:

```
The current base URI in force is as defined in the
W3C <ref  target="http://www.w3.org/TR/xmlbase/">XML
Base</ref> recommendation.
```

This example points explicitly to a location on the Web, accessible via HTTP. Suppose however that we wish to access a document stored locally in a file. Again we will supply an absolute URI reference, but this time using a different protocol:

```
This Debian package is distributed under the terms
of the <ref
   target="file:///usr/share/common-licenses/GPL-2">GNU General Public License</ref>.
```

In the following example, we use a relative URI reference to point to a local document:

```
<figure  rend="float fullpage">
  <graphic  url="Images/compic.png"/>
  <figDesc>The figure shows the page from the <title>Orbis
        pictus</title> of Comenius which is discussed in the text.</figDesc>
</figure>
```

Since no **xml:base** is specified here, the location of the resource Figures/compic.png is determined relative to the resource indicated by the current base URI, which is the current document.

In the following example, however, we first change the current base URI by setting a new value for **xml:base**. The resource required is then identified by means of a relative URI:

```
<div  type="chap"  xml:base="http://classics.mit.edu/">
  <head>On Ancient Persian Manners</head>
  <p>In the very first story of <ref  target="Sadi/gulistan.2.i.html">
     <title>The Gulistan of
           Sa'di</title>
   </ref>,
    Sa'di relates moral advice worthy of Miss Minners ...</p>
<!-- ... -->
</div>
```

As noted above, the current base URI is found on the nearest ancestor. This provides a useful way of abbreviating URIs within a given scope:

```
<body>
  <div  n="A">
    <p>The base URI here is the current document.  A URI such as
    <code>a.xml</code> is equivalent to
```

```
    <code>./a.xml</code>.</p>
  </div>
  <div  n="B"  xml:base="http://www.example.org/">
    <p>The base URI here is
    <code>http://www.example.org/</code>.  A
        URI such as <code>a.xml</code> is equivalent to
    <code>http://www.example.org/a.xml</code>.</p>
  </div>
  <div  n="C"  xml:base="ftp://ftp.example.net/mirror/">
    <p>The base URI here is
    <code>ftp://ftp.example.net/mirror/</code>.  A URI such
        as
    <code>a.xml</code> is equivalent to
    <code>ftp://ftp.example.net/mirror/a.xml</code>.</p>
  </div>
  <div  n="D">
    <p>The base URI here is the current document.  A URI such as
    <code>a.xml</code> is equivalent to
    <code>./a.xml</code>.</p>
  </div>
</body>
```

16.2.2 Pointing Locally

Because the default base URI is the current document, a pointer that is specified as a bare name fragment identifier alone acts as a pointer to an element in the current document, as in the following example.

```
<div  type="section"  xml:id="sect106">
<!-- ... -->
</div>
<div  type="section"  n="107"  xml:id="sect107">
  <head>Limitations on exclusive rights: Fair use</head>
  <p>Notwithstanding the provisions of
  <ref  target="#sect106">section 106</ref>, the fair use of a
     copyrighted work, including such use by reproduction in copies
     or phonorecords or by any other means specified by that section,
     for purposes such as criticism, comment, news reporting,
     teaching (including multiple copies for classroom use),
     scholarship, or research, is not an infringement of copyright.
     In determining whether the use made of a work in any particular
     case is a fair use the factors to be considered shall
     include –
  <list  type="simple">
     <item  n="(1)">the purpose and character of the use, including
           whether such use is of a commercial nature or is for nonprofit
           educational purposes;</item>
     <item  n="(2)">the nature of the copyrighted work;</item>
     <item  n="(3)">the amount and substantiality of the portion
           used in relation to the copyrighted work as a whole;
           and</item>
     <item  n="(4)">the effect of the use upon the potential market
           for or value of the copyrighted work.</item>
    </list>
     The fact that a work is unpublished shall not itself bar a
     finding of fair use if such finding is made upon consideration
```

```
    of all the above factors.</p>
</div>
```

Source: [200]

This method of pointing, by referring to the **xml:id** of the target element as a bare name only (e.g., **#sect106**) is the simplest and often the best approach where it can be applied, i.e. where both the source element and target element are in the same XML document, and where the target element carries an identifier. It is the method used extensively in previous sections of this chapter and elsewhere in these Guidelines.

16.2.3 W3C element() Scheme

If elements are not directly addressable by means of an identifier, because no identifier was originally given to them and the document cannot be modified to add one, they may still be pointed to by means of their position in the XML element tree. This method of pointing uses the element() scheme defined by the World Wide Web Consortium (Grosso et al, 2003). In this scheme, an element may be identified by stepwise navigation using a slash-separated list of child element numbers. For each step the integer `n` locates the **n**th child element of the previously located element. Thus a pointer such as `<ptr target="foo.xml#element(/1/4)"/>` indicates the fourth child element starting from the root element of the document indicated by the URI foo.xml.

For example, the following pointer selects one of Shakespeare's most famous lines:

```
<ref
   target="http://www.cs.mu.oz.au/621/2003project/hamlet.xml#element(/1/8/2/25/2)">2B|^2B…</ref>
```

The URI in this example references an XML resource assumed to be available via the HTTP protocol on the Web; within that file, the specified element() scheme is used to select 'the first (root-level) element's 8th child element's 2nd child element's 25th child element's 2nd child element'. This is equivalent to the XPath specification `/*[1]/*[8]/*[2]/*[25]/*[2]`.

Rather than specifying a full path starting from the document root, it is also possible in this pointer scheme to specify as starting point any element which carries a value for its **xml:id** attribute, supplying a unique identifier for it. In this case the identifier is prefixed to the location path. For example, we can point more economically to the same line of Hamlet in a different digital version of the play which provides identifiers for the individual scenes:

```
<div
   xml:base="/Users/martin/Documents/c5/namelessShakespeare.xml">
  <p>
    <ptr  target="#element(sha-ham301/22/2)"/>
  </p>
</div>
```

Here the identifier **sha-ham301** is the identifier for the element containing Act III, Scene I of Hamlet. The second child of the 22nd child of this element contains the desired <l> element. This is equivalent to the XPath specfication `id(sha-ham301)/*[22]/*[2]`.

As noted above, we could also point directly to this line if it had an identifier of its own. In another digital edition of Shakespeare, based on the first folio, each line is given an identifier based on its 'through line number'. Our pointer to this line can now be represented simply as **<ptr target="#element(Ham01245)"/>**, or even more simply as **<ptr target="#Ham01245"/>**. The notation **<ptr target="#xxx"/>** is a convenient abbreviation for **<ptr target="#element(xxx)"/>**. This method requires, of course, that the 'Through Line Number' is supplied as the value of an **xml:id** attribute on each line, and must therefore be unique within each document. In section *16.2.5. Canonical References* we discuss a method of pointing to the line which does not have this requirement.

16.2.4 TEI XPointer Schemes

The pointing scheme described in this chapter is one of a number of such schemes envisaged by the W3C, which together constitute a framework for addressing data within XML documents, known as the XPointer Framework (Grosso et al 2003). This framework permits the definition of many other named addressing methods, each of which is known as an XPointer Scheme. The W3C has predefined a set of such schemes, and maintains a register for their expansion. The element() scheme described above is one such scheme, defined by the W3C, and widely implemented by XML processing systems.

Another important scheme, also defined by the W3C, and recommended by these Guidelines is the xpath1() pointer scheme, which allows for any part of an XML structure to be selected using the syntax defined by the XPath specification. This is further discussed below, *16.2.4.2. xpath1(Expr)*. These Guidelines also define five other pointer schemes, which provide access to parts of an XML document such as points within data content or stretches of data content. These additional TEI pointer schemes are defined in sections *16.2.4.3. left(pointer) and right(pointer)* to *16.2.4.6. match(pointer, string [, index])* below.

16.2.4.1 Introduction to TEI Pointers

Before discussing the TEI pointer schemes, we introduce slightly more formally the terminology used to define them. So far, we have discussed only ways of pointing at components of the XML information set node such as elements and attributes. However, there is often a need in text analysis to address additional types of location such as the 'point' locations *between* nodes, and 'ranges' that may arbitrarily cross the boundaries of nodes in a document. The content of an XML document is organized sequentially as well as hierarchically, and it therefore makes sense to consider ranges of characters within it independently of the nodes to which they belong, for example when making a selection in a text editor. For processing purposes, such a range is best defined by the pair of points at its start and end. It is often useful to think of pointer schemes as analogous to query functions that return nodes in the XML information set (the DOM tree) of an XML document, as in the case of the element and xpath pointer schemes discussed so far, but this is not invariably the case. A point is adjacent to one or two nodes, but is not a node itself, while a range may not even overlap with any complete node in the DOM tree.

The TEI pointer scheme thus distinguishes the following kinds of object:

Node A node represents a single item in the XML information set for a document. For pointing purposes, the only nodes that are of interest are Text Nodes, Element Nodes, and Attribute nodes.

Node Set A node set is a set of nodes in the XML information set of a document. In TEI Pointing applications, node sets are only allowed as the result of resolving a URI when multiple URIs would have been allowed where it appears, i.e. in attributes which are declared as permitting two or more data.pointer values as opposed to only one. As the name 'set,' implies, the individual items in a node set are not ordered, and no assumptions about relative ordering of items in a node set should be made.

Point A Point represents a point between nodes in a document. Every point is adjacent to either characters or elements, and never to another point. In fact, in the character representation of an XML document, every position between data characters, start-tags or end-tags is a point, and there are no other points. If one treats all character content as if it were broken into single-character text-nodes, every point is definable as either

- the point preceding a node, and if that node has a predecessor in document order, then it is the same as the point following that predecessor; or
- the point following a node, and if that node has a successor in document order, then it is the same as the point preceding that successor.

Range A Range is defined as the portion of a document between two points. Since points may occur anywhere within the document, ranges do not correspond directly to nodes or to node sets. A range may overlap the contents of a node either completely or partially.

The TEI has registered the following five pointer schemes:

xpath1() Addresses a node or nodeset using the XPath syntax. (*16.2.4.2. xpath1(Expr)*)

left() and right() addresses the point before (left) or after (right) a node or node set (*16.2.4.3. left(pointer) and right(pointer)*)

range() addresses the range between two points (*16.2.4.4. range(pointer1, pointer2)*)

string-range() addresses a range of a specified length starting from a specified point (*16.2.4.4. range(pointer1, pointer2)*)

match() addresses a range which matches a specified string within a node (*16.2.4.6. match(pointer, string [, index])*)

The xpath1() scheme refers to the existing XPath specification which is adopted without modification or extension.

The other five schemes overlap in functionality with a W3C draft specification known as the XPointer scheme draft, but are individually much simpler. At the time of this writing, there is no current or scheduled activity at the W3C towards revising this draft or issuing it as a recommendation.

16.2.4.2 xpath1(Expr)

The xpath1() scheme locates a node or node set within an XML Information Set. The single argument Expr is an XPath Expr as defined in the W3C XPath 1 Recommendation. The node or node set resulting from evaluating the XPath is the reference of an address using the xpath1() scheme. For example, the following example selects the first paragraph of the <ftnote> element with **id** of **fn6** of a paper that discusses XPointers.

```
<ptr
   target="http://tinyurl.com/267z62/xml/2004/Thompson01/EML2004Thompson01.xml#xpath1(//ftnote[@id='fn6']/para[1])"/>
```

When a URI reference is specified as the value of an attribute declared as a single data.pointer value, the result must be a single node, and it is an error if the result is a node set. When the URI reference is specified as the value of an attribute declared to permit two or more data.pointer values, each node in the node set is treated as if it were the result of a separate URI reference.

When an xpath is interpreted by a TEI processor, the information set of the referenced document is interpreted without any additional information supplied by any schema processing that may or may not be present. In particular this means that no whitespace normalization is applied to a document before the xpath is interpreted.

This pointer scheme allows easy, direct use of the most widely-implemented XML query method. It is probably the most robust pointing mechanism for the common situation of selecting an XML element or its contents where an **xml:id** is not present. The ability to use element names and attribute names and values makes xpath1() pointers more robust than the other mechanisms discussed in this section

even if the designated document changes. For durability in the presence of editing, use of **xml:id** is always recommended when possible.

16.2.4.3 left(pointer) and right(pointer)

The left() (right()) scheme locates the point immediately preceding (following) its argument. The single pointer argument to left() or right() is treated like a fragment identifier itself, and must be a bare name or XPointer pointer. The designation of this argument is resolved with respect to the base URI in effect for the left() or right() according to the normal rules.[5] Most pointer schemes return nodes or ranges rather than points; the possibilities for left() and right() pointer schemes are as follows:

A Node When pointer resolves to a node, the point designated is the point immediately preceding (left()) or following (right()) the node.

A Node Set When pointer resolves to a node set, the point designated is the point preceding the first element of the set (left()) or following the last element of the set (right())

A range When pointer resolves to a range, the point designated is the point designating the start (left()) or end (right()) of the range.

A Point When pointer resolves to a point, that point is the result. The pointer schemes left() and right() make no change when given a point as argument.

The following example points to the spot immediately following the last character of the element found by walking down the document tree to the 6th child of the 3rd child of the 3rd child of the 1st child of the root element. In this case, the path takes us to a <postcode> node which contains the string '20850', so the point being pointed to is that following the '0' character at the end of the element content.

```
<p
   xml:base="http://www.mulberrytech.com/Extreme/Proceedings/xml/2002/">
  <ptr
     target="Usdin01/EML2002Usdin01.xml#right(element(/1/1/3/3/6))"/>
</p>
```

16.2.4.4 range(pointer1, pointer2)

The range() scheme locates a range between two points in an XML information set. The two pointer arguments to range() locate the boundaries of the range by two points, and are interpreted as fragment identifiers. The parameters pointer1 and pointer2 are XPointers themselves, and are resolved according to the rules specified in the definition of the pointer scheme they use.[6] Most pointer schemes return nodes or ranges rather than points; the possibilities for range() pointer schemes are as follows:

A Node When pointer1 resolves to a node, the starting point of the range is the point immediately preceding the node. When pointer2 resolves to a node, the ending point of the range is the point immediately following the node. It is an error if the ending point precedes the starting point of a range.

A range When pointer1 resolves to a range R, the starting point of the result range is the same as the starting point of R. When pointer2 resolves to a range R, the ending point of the result range is the ending point of R.

A Point When pointer1 resolves to a point, that point is the start of the range. When pointer2 resolves to a point, that point is the end of the range.

[5] Like other XPointer schemes, bare names (i.e. values of **xml:id** references) are permitted as pointer arguments to all TEI-defined XPointer pointer scheme parameters.

[6] Bare names (i.e., **xml:id** values), like other Xpointer schemes, are permitted as range() parameters.

16.2.4.5)

]string-range(pointer, offset [, length])

The string-range() scheme locates a range based on character positions.

While string-range endpoints are points adjacent to character positions, they must be designated by the characters to which they are adjacent, in the same way that the nodes corresponding to XML elements are. This avoids ambiguity about which point between two characters is indicated when characters are interrupted by markup.

The pointer argument to string-range() designates a node or a range within which a string is to be located. No string range, even an empty one, can be defined by a string-range() if pointer has the empty string as string value.

Every string-range is defined based on an 'origin character'. The origin is numbered 0, and designates the first character of the string-value of pointer. The offset is a character index relative to the origin; the start of the resulting range is the position designated by the sum of the origin and offset.

If length is specified, the end of the range is at a point adjacent to the character designated by the origin added to the offset and length. If the offset is negative, or length is sufficiently large, a string-range can designate characters outside the string-value of the intitial pointer. In this case, characters are located using the string-value of the entire document. It is also legal for length plus the origin to exceed the length of the string-value of the document by one, in order to accommodate ranges that include the last character of a document.

If length is not specified, it defaults to the value 1, and the string range contains one character. If it is specified as 0, the zero-length range is interpreted as the point immediately preceding the origin character or offset character if there is one.

16.2.4.6)

]match(pointer, string [, index])

The match scheme designates the result of a literal match of the argument string within the string-value of the pointer argument. The result is a range from the first matching character to the last. It is an error if there is no matching string. A match may not extend outside the range corresponding to the string value of pointer.

The index argument is an integer greater than or equal to 1, specifying which match should be chosen when there is more than one match within the string-value of pointer. If no index is provided, the default value is 1, indicating the first match found.

16.2.5 Canonical References

By 'canonical' reference we mean any means of pointing into documents, specific to a community or corpus. For example, biblical scholars might understand 'Matt 5:7' to mean 'the book called *Matthew*, chapter 5, verse 7.' They might then wish to translate the string 'Matt 5:7' into a pointer into a TEI-encoded document, selecting the element which corresponds to the seventh element within the fifth element within the element with the **n** attribute valued 'Matt.'

Several elements in the TEI scheme (<gloss>, <ptr>, <ref>, and <term>) bear a special attribute, **cRef**, just for this purpose. Using the system described in this section, an encoder may specify references to canonical works in a discipline-familiar format, and expect software to derive a complete URI from it. The value of the **cRef** attribute is processed as described in this section, and the resulting URI reference is treated as if it were the value of the **target** attribute. The **cRef** and **target** attributes are mutually exclusive: only one or the other may be specified on any given occurrence of an element.

For the **cRef** attribute to function as required, a mechanism is needed to define the mapping between (for example) 'the book called *Matt*' and the part of the XML structure which corresponds with it. This is provided by the <refsDecl> element in the TEI Header, which contains an algorithm for translating a canonical reference string (like **Matt 5:7**) into a URI such as `#xpath1(//div[@n='Matt']/div[5]/div[7]`.

The <refsDecl> element is described in section *2.3.5. The Reference System Declaration*; the following example is discussed in more detail below in section *16.2.5.1. Worked Example*.

```
<refsDecl xml:id="biblical">
  <cRefPattern
     matchPattern="(.+) (.+):(.+)"
     replacementPattern="#xpath1(//div[@n='$1']/div[$2]/div[$3])">
    <p>This pointer pattern extracts and references the <q>book,</q>
      <q>chapter,</q> and <q>verse</q> parts of a biblical reference.</p>
  </cRefPattern>
  <cRefPattern matchPattern="(.+) (.+)"
     replacementPattern="#xpath1(//div[@n='$1']/div[$2])">
    <p>This pointer pattern extracts and references the <q>book</q> and
    <q>chapter</q> parts of a biblical reference.</p>
  </cRefPattern>
  <cRefPattern matchPattern="(.+)"
     replacementPattern="#xpath1(//div[@n='$1'])">
    <p>This pointer pattern extracts and references just the <q>book</q>
       part of a biblical reference.</p>
  </cRefPattern>
</refsDecl>
```

When an application encounters a canonical reference as the value of **cRef** attribute, it follows a sequence of specific steps to transform it into a URI reference.

1. Ascertain the correct <refsDecl> following the rules summarized in section *15.3.3. Summary*.
2. For each <cRefPattern> element encountered in the appropriate <refsDecl>, in the order encountered:
   (a) match the value of **cRef** to the regular expression found as the value of the **matchPattern** attribute
   (b) if the **cRef** value matches, take the value of the **replacementPattern** attribute and substitute the back references ($1, $2, etc.) with the corresponding matched substrings
   (c) the result is taken as if it were a relative or absolute URI reference specified on the **target** attribute; i.e., it should be used as is or combined with the current **xml:base** value as usual
   (d) no further processing of this **cRef** against the <refsDecl> should take place
   (e) if, however, the **cRef** value does not match the regular expression specified on **matchPattern** attribute, proceed to the next <cRefPattern>
3. If all the <cRefPattern> elements are examined in turn and none matches, the pointer fails.

The regular expression language used as the value of the **matchPattern** attribute is that used for the pattern facet of the World Wide Web Consortium's XML Schema Language in an Appendix to XML Schema Part 2.[7] The value of the **replacementPattern** attribute is simply a string, except that occurences of '$1' through '$9' are replaced by the corresponding substring match. Note that since a maximum of nine substring matches are permitted, the string '$18' means 'the value of the first matched substring followed by the character '8" as opposed to 'the eighteenth matched substring'. If there is a need for an actual string including a dollar sign followed by a digit that is not supposed to be replaced, the dollar sign should be written as `%24`.

[7] As always seems to be the case, no two regular expression languages are precisely the same. For those used to Perl regular expressions, be warned that while in Perl the pattern `tei` matches any string that contains *tei*, in the W3C language it only matches the string 'tei'.

16.2.5.1 Worked Example

Let us presume that with the example <refsDecl> above, an application comes across a **cRef** value of **Matt 5:7** inside a which has an **xml:base** of **http://www.example.org/resources/books/Bible.xml**. The application would first apply the regular expression `(.+) (.+):(.+)` to 'Matt 5:7'. This regular expression would successfully match. The first matched substring would be 'Matt', the second '5', and the third '7'. The application would then apply these substrings to the pattern `#xpath1(//div[@n='$1']/div[$2]/div[$3])`, producing `#xpath1(//div[@n='Matt']/div[5]/div[7])`. It would append this to the **xml:base** in force, thus generating the complete URI Reference `http://www.example.org/resources/books/Bible.xml#xpath1(//div[@n='Matt']/div[5]/div[7])`.

If, however, the input string had been 'Matt 5', the first regular expression would not have matched. The application would have then tried the second, `(.+) (.+)`, producing a successful match, and the matched substrings 'Matt' and '5'. It would then have substituted those matched substrings into the pattern `#xpath1(//div[@n='$1']/div[$2])` to produce a fragment identifier, which when appended to the **xml:base** in force produces the absolute URI reference `http://www.example.org/resources/books/Bible.xml#xpath1(//div[@n='Matt']/div[5])`.

If the input string had been 'Matt', neither the first nor the second regular expressions would have successfully matched. The application would have then tried the third, `(.+)`, producing the matched substring 'Matt', and the URI Reference `http://www.example.org/resources/books/Bible.xml#xpath1(//div[@n='Matt'])`.

It is an error to reference more matched substrings than are produced by the regular expression. For example:

```
<cRefPattern
   matchPattern="(.+) (.+):(.+)"
   replacementPattern="//div[@n='$1']/div[$2]/div[$3]/p[$4]"/>
```

would produce an error, since only three matched substrings would have been produced, but a fourth (`$4`) was referenced.

It is quite reasonable to believe that encoders would actually prefer much more precise regular expressions than those used as examples above. E.g., `^\s*([1-9]?[A-Z][a-z]+)\s+([1-9][0-9]?[0-9]?):([1-9][0-9]?)\s*$`.

16.2.5.2 Complete and Partial URI Examples

In the above example, the value of **cRef** was used to generate a Fragment Identifier, which in turn was used to generate a complete URI. The complete URI could be generated directly, as in the following example.

```
<refsDecl xml:id="USC">
  <cRefPattern
     matchPattern="([0-9][0-9])\s*U\.?S\.?C\.?\s*[Cc](h(\.|ap(ter|\.)?)?)?\s*([1-9][0-9]*)"
     replacementPattern="http://uscode.house.gov/download/pls/$1C$5.txt">
    <p>Matches most standard references to particular
        chapters of the United States Code, e.g.
    <val>11USCC7</val>, <val>17 U.S.C. Chapter 3</val>, or
    <val>14 USC Ch. 5</val>. Note that a leading zero is
        required for the title (must be two digits), but is not
        permitted for the chapter number.</p>
  </cRefPattern>
  <cRefPattern
     matchPattern="([0-9][0-9])\s*U\.?S\.?C\.?\s*[Pp](re(lim(inary)?)?)?\s*[Mm](at(erial)?)?"
     replacementPattern="http://uscode.house.gov/download/pls/$1T.txt">
    <p>Matches references to the preliminary material for a
```

```
      given title, e.g. <val>11USCP</val>, <val>17 U.S.C.
         Prelim Mat</val>, or <val>14 USC pm</val>.</p>
  </cRefPattern>
  <cRefPattern
     matchPattern="([0-9][0-9])\s*U\.?S\.?C\.?\s*[Aa](ppend(ix)?)?"
     replacementPattern="http://uscode.house.gov/download/pls/$1A.txt">
    <p>Matches references to the appendix of a given tile,
       e.g. <val>05USCA</val>, <val>11 U.S.C. Appendix</val>,
       or <val>18 USC Append</val>.</p>
  </cRefPattern>
</refsDecl>
<!-- ... -->
<p>The example in section <ptr  target="#SABN"/> is taken
from <ref  cRef="17 USC Ch 1">Subject Matter and Scope of
    Copyright</ref>.</p>
```

16.2.5.3 Miscellaneous Usages

Canonical reference pointers are intended for use by TEI encoders. However, this specification might be useful to the development of a process for recognizing canonical references in non-TEI documents (such as plain text documents), possibly as part of their conversion to TEI.

16.3 Blocks, Segments, and Anchors

In this section, we discuss three general purposes elements which may be used to mark and categorize both a span of text and a point within one. These elements have several uses, most notably to provide elements which can be given identifiers for use when aligning or linking to parts of a document, as discussed elsewhere in this chapter. They also provide a convenient way of extending the semantics of the TEI markup scheme in a theory-neutral manner, by providing for two neutral or 'anonymous' elements to which the encoder can add any meaning not supplied by other TEI defined elements.

<anchor/> (anchor point) attaches an identifier to a point within a text, whether or not it corresponds with a textual element.

<ab> (anonymous block) contains any arbitrary component-level unit of text, acting as an anonymous container for phrase or inter level elements analogous to, but without the semantic baggage of, a paragraph.

@**part** specifies whether or not the block is complete.

<seg> (arbitrary segment) represents any segmentation of text below the 'chunk' level.

@**subtype** provides a sub-categorization of the segment marked.

The elements <anchor> and <ab> are both members of the class att.typed, from which they inherit the following attributes:

att.typed provides attributes which can be used to classify or subclassify elements in any way.

@**type** characterizes the element in some sense, using any convenient classification scheme or typology.

@**subtype** provides a sub-categorization of the element, if needed

The <seg> element is a member of the classes att.segLike from which it inherits the following attributes:

att.segLike provides attributes for elements used for arbitrary segmentation.

@**type** characterizes the type of segment.

@**function** characterizes the function of the segment.

@part specifies whether or not the segment is fragmented by some other structural element, for example a clause which is divided between two or more sentences.

The <anchor> element may be thought of as an empty <seg>, or as an artifice enabling an identifier to be attached to any position in a text. Like the <milestone> element discussed in section *3.10. Reference Systems*, it is useful where multiple views of a document are to be combined, for example, when a logical view based on paragraphs or verse lines is to be mapped on to a physical view based on manuscript lines. Like those elements, it is a member of the class model.global and can therefore appear anywhere within a document when the module defined by this chapter is included in a schema. Unlike the other elements in its class, the <anchor> element is primarily intended to mark an arbitrary point used for alignment, or as the target of a spanning element such as those discussed in section *11.3.4. Additions and Deletions*, rather than as a means of marking segment boundaries for some arbitrary segmentation of a text.

For example, suppose that we wish to mark the end of the fifth word following each occurrence of some term in a particular text, perhaps to assist with some collocational analysis. This can most easily be done with the help of the <anchor> element, as follows:

```
English language. Except for not very<anchor  xml:id="eng1"/>
English at all at the time<anchor  xml:id="eng2"/>
English was still full of flaws<anchor  xml:id="eng3"/>
English. This was revised by young
<anchor  xml:id="eng4"/>
```

In section *16.4.1. Correspondence* we discuss ways in which these <anchor> points might be used to represent an alignment such as one might get in a keyword-in-context concordance.

The <seg> element may be used at the encoder's discretion to mark almost any segment of the text of interest for processing. One use of the element is to mark text features for which no appropriate markup is otherwise defined, i.e. as a simple extension mechanism. Another use is to provide an identifier for some segment which is to be pointed at by some other element, i.e. to provide a target, or a part of a target, for a <ptr> or other similar element.

Several examples of uses for the <seg> element are provided elsewhere in these Guidelines. For example:

- as a means of marking segments significant in a metrical or rhyming analysis (see section *6.3. Rhyme and Metrical Analysis*)
- as a means of marking typographic lines in drama (see section *7.2. The Body of a Performance Text*) or title pages (see section *4.6. Title Pages*)
- as a means of marking prosody- or pause-defined units in transcribed speech (see section *8.4.1. Segmentation*)
- as a means of marking linguistic or other analyses in a theory-neutral manner (see chapter *17. Simple Analytic Mechanisms* passim)

In the following simple example, the <seg> element simply delimits the extent of a stutter, a textual feature for which no element is provided in these Guidelines.

```
<q>Don't say <q>
   <seg  type="stutter">I-I-I</seg>'m afraid,</q> Melvin, just say <q>I'm
    afraid.</q>
</q>
```

Source: [181]

The <seg> element is particularly useful for the markup of linguistically significant constituents such as the phrases that may be the output of an automatic parsing system. This example also demonstrates the use of the **xml:id** attribute to carry an identifier which other parts of a document may use to point to, or align with:

```
<seg xml:id="bl0034" type="sentence">
  <seg xml:id="bl0034.1" type="phrase">Literate and illiterate speech</seg>
  <seg xml:id="bl0034.2" type="phrase">in a language like English</seg>
  <seg xml:id="bl0034.3" type="phrase">are plainly different.</seg>
</seg>
```

Source: [17]

As the above example shows, <seg> elements may be nested directly within one another, to any degree of analysis considered appropriate. This is taken a little further in the following example, where the **type** and **subtype** attributes have been used to further categorise each word of the sentence (the **xml:id** attributes have been removed to reduce the complexity of the example):

```
<seg type="sentence" subtype="declarative">
  <seg type="phrase" subtype="noun">
    <seg type="word" subtype="adjective">Literate</seg>
    <seg type="word" subtype="conjunction">and</seg>
    <seg type="word" subtype="adjective">illiterate</seg>
    <seg type="word" subtype="noun">speech</seg>
  </seg>
  <seg type="phrase" subtype="preposition">
    <seg type="word" subtype="preposition">in</seg>
    <seg type="word" subtype="article">a</seg>
    <seg type="word" subtype="noun">language</seg>
    <seg type="word" subtype="preposition">like</seg>
    <seg type="word" subtype="noun">English</seg>
  </seg>
  <seg type="phrase" subtype="verb">
    <seg type="word" subtype="verb">are</seg>
    <seg type="word" subtype="adverb">plainly</seg>
    <seg type="word" subtype="adjective">different</seg>
  </seg>
  <seg type="punct">.</seg>
</seg>
```

(The example values shown are chosen for simplicity of comprehension, rather than verisimilitude). It should also be noted that specialized segment elements are defined in section *17.1. Linguistic Segment Categories* to facilitate this particular kind of analysis. These allow for the explicit markup of units called s-units, clauses, phrases, words, morphemes, and characters, which may be felt preferable to the more generic approach typified by use of the <seg> element. Using these, the first phrase above might be encoded simply as

```
<phr type="noun">
  <w type="adjective">Literate</w>
  <w type="conjunction">and</w>
  <w type="adjective">illiterate</w>
  <w type="noun">speech</w>
</phr>
```

Note the way in which the **type** attribute of these specialized elements now carries the value carried by the **subtype** attribute of the more general <seg> element. For an analysis not using these traditional linguistic categories however, the <seg> element provides a simple but powerful mechanism.

In language corpora and similar material, the <seg> element may be used to provide an end-to-end segmentation as an alternative to the more specific <s> element proposed in chapter *17.1. Linguistic Segment Categories* for the markup of orthographic sentences, or s-units. However, it may be more useful to use the <s> element for this purpose, since this means that the <seg> element can then be used to mark both features within s-units and segments composed of s-units, as in the following example:[8]

```
<seg xml:id="s1s3" type="narrative_unit">
  <s xml:id="s1">Sigmund, the <seg type="patronymic">son of Volsung</seg>,
    was a king in Frankish country.</s>
  <s xml:id="s2">Sinfiotli was the eldest of his sons.</s>
  <s xml:id="s3"> ... </s>
</seg>
```

Like other elements, the <seg> tag must be properly enclosed within other elements. Thus, a single <seg> element can be used to group together words in different sentences only if the sentences are not themselves tagged. The first of the following two encodings is legal, but the second is not.

```
Give me <seg type="phrase">a dozen. Or two or three.</seg>
```

```
    <!-- Illegal! -->
    <s>Give me <seg type="phrase">a dozen.</s>
    <s>Or two or three.</s></seg>
```

The **part** attribute may be used as one simple method of overcoming this restriction:

```
<s>Give me <seg type="phrase" part="I">a dozen.</seg>
</s>
<s>
  <seg part="F">Or two or three.</seg>
</s>
```

Another solution is to use the <join> element discussed in section *16.7. Aggregation*; this requires that each of the <seg> elements be given an identifier. For further discussion of this generic encoding problem, see also chapter *20. Non-hierarchical Structures*.

The <seg> element has the same content as a paragraph in prose: it can therefore be used to group together consecutive sequences of model.inter class elements, such as lists, quotations, notes, stage directions, etc. as well as to contain sequences of phrase-level elements. It cannot however be used to group together sequences of paragraphs or similar text units such as verse lines; for this purpose, the encoder should use intermediate pointers, as described in section *16.1.4. Intermediate Pointers* or the methods described in section *16.7. Aggregation*. It is particularly important that the encoder provide a clear description of the principles by which a text has been segmented, and the way in which that segmentation is represented. This should include a description of the method used and the significance of any categorization codes. The description should be provided as a series of paragraphs within the <segmentation> element of the encoding description in the TEI header, as described in section *2.3.3. The Editorial Practices Declaration*.

The <seg> element may also be used to encode simultaneous or mutually exclusive variants of a text when the more special purpose elements for simple editorial changes, abbreviation and expansion, addition and

[8] See section *17.3. Spans and Interpretations*, where the text from which this fragment is taken is analyzed.

deletion, or for a critical apparatus are not appropriate. In these circumstances, one <seg> is encoded for each possible variant, and the set of them is enclosed in a <choice> element.

For example, if one were writing dual-platform instructions for installation of software, it might be useful to use <seg> to record platform-specific pieces of mutually exclusive text.

```
…pressing <choice>
  <seg  type="platform"  subtype="Mac">option</seg>
  <seg  type="platform"  subtype="PC">alt</seg>
</choice>-f will …
```

Elsewhere in this chapter we provide a number of examples where the <seg> element is used simply to provide an element to which an identifier may be attached, for example so that another segment may be linked or related to it in some way.

The <ab> (anonymous block) element performs a similar function to that of the <seg> element, but is used for portions of the text which occur not within paragraphs or other component-level elements, but at the component level themselves. It is therefore a member of the model.pLike class.

The <ab> element may be used, for example, to tag the canonical verse divisions of Biblical texts:

```
<div1  n="Gen"  type="book">
  <head>The First Book of Moses, Called</head>
  <head  type="main">Genesis</head>
  <div2  n="1"  type="chapter">
    <ab  n="1">In the beginning God created the heaven and the
        earth.</ab>
    <ab  n="2">And the earth was without form, and void; and darkness
    <hi>was</hi> upon the face of the deep. And the Spirit of God
        moved upon the face of the waters.</ab>
    <ab  n="3">And God said, Let there be light: and there was
        light.</ab>
  </div2>
</div1>
```

Source: [83]

In other cases, where the text clearly indicates paragraph divisions containing one or more verses, the <p> element may be used to tag the paragraphs, and the <seg> element used to subdivide them. The <ab> element is provided as an alternative to the <p> element; it may *not* be used within paragraphs. The <seg> element, by contrast, may appear only within and not between paragraphs (or anonymous block elements).

```
<div1  n="Gen"  type="book">
  <head>Das Erste Buch Mose.</head>
  <div2  n="1"  type="chapter">
    <p>
      <seg  n="1">Am Anfang schuff Gott Himel vnd Erden.</seg>
      <seg  n="2">Vnd die Erde war wüst vnd leer / vnd es war
            finster auff der Tieffe / Vnd der Geist Gottes schwebet auff
            dem Wasser.</seg>
    </p>
    <p>
      <seg  n="3">Vnd Gott sprach / Es werde Liecht / Vnd es ward
            Liecht.</seg>
    </p>
  </div2>
</div1>
```

Source: [132]

The <ab> element is also useful for marking dramatic speeches when it is not clear whether the speech is to be regarded as prose or verse. If, for example, an encoder does not wish to express an opinion as to whether the opening lines of Shakespeare's *The Tempest* are to be regarded as prose or as verse, they might be tagged as follows:

```
<div1  n="I"  type="act">
  <div2  n="1"  type="scene">
    <head  rend="italic">Actus primus, Scena prima.</head>
    <stage  rend="italic"  type="setting"> A tempestuous noise of
        Thunder and Lightning heard:
        Enter a Ship-master, and a Boteswaine.</stage>
    <sp>
      <speaker>Master.</speaker>
      <ab>Bote-swaine.</ab>
    </sp>
    <sp>
      <speaker>Botes.</speaker>
      <ab>Heere Master: What cheere?</ab>
    </sp>
    <sp>
      <speaker>Mast.</speaker>
      <ab>Good: Speake to th' Mariners: fall too't, yarely,
            or we run our selues a ground, bestirre, bestirre.
      <stage  type="move">Exit.</stage>
      </ab>
    </sp>
    <stage  type="move">Enter Mariners.</stage>
    <sp>
      <speaker>Botes.</speaker>
      <ab>Heigh my hearts, cheerely, cheerely my harts: yare, yare:
            Take in the toppe-sale: Tend to th' Masters whistle: Blow
            till thou burst thy winde, if roome e-nough.</ab>
    </sp>
  </div2>
</div1>
```

Source: [177]

See further *3.12.2. Core Tags for Drama* and *7.2.4. Speech Contents*.

16.4 Correspondence and Alignment

In this section we introduce the notions of correspondence, expressed by the **corresp** attribute, and of alignment, which is a special kind of correspondence involving an ordered set of correspondences. Both cases may be represented using the <link> and <linkGrp> elements introduced in section *16.1. Links*. We also discuss the special case of alignment in time or synchronization, for which special purpose elements are proposed in section *16.5. Synchronization*.

16.4.1 Correspondence

A common requirement in text analysis is to represent correspondences between two or more parts of a single document, or between places in different documents. Provided that explicit elements are available to represent the parts or places to be linked, then the global linking attribute **corresp** may be used to encode such correspondence, once it has been identified.

att.global.linking defines a set of attributes for hypertext and other linking, which are enabled for all elements when the additional tag set for linking is selected.

@**corresp** (corresponds) points to elements that correspond to the current element in some way.

This is one of the attributes made available by the mechanism described in the introduction to this chapter (*16. Linking, Segmentation, and Alignment*). Correspondence can also be expressed by means of the <link> element introduced in section *16.1. Links*.

Where the correspondence is between *spans*, the <seg> element should be used, if no other element is available. Where the correspondence is between *points*, the <anchor> element should be used, if no other element is available.

The use of the **corresp** attribute with spans of content is illustrated by the following example:

```
<title xml:id="SHIRLEY">Shirley</title>, which made
its Friday night debut only a month ago, was
not listed on <name xml:id="NBC">NBC</name>'s new schedule,
although <seg corresp="#NBC" xml:id="NETWORK">the network</seg>
says <seg corresp="#SHIRLEY" xml:id="SHOW">the show</seg>
still is being considered.
```

Source: [122]

Here the anaphoric phrases *the network* and *the show* have been associated directly with the elements to which they refer by means of **corresp** attributes. This mechanism is simple to apply, but has the drawback that it is not possible to specify more exactly what kind of correspondence is intended. Where this attribute is used, therefore, encoders are encouraged to specify their intent in the associated encoding declarations in the TEI Header.

Essentially, what the **corresp** attribute does is to specify that the element that has the attribute and the element(s) the attribute points to are doubly linked.[9] Therefore, we can also use the <link> and <linkGrp> elements defined in section *16.1. Links* to indicate correspondence among elements. Moreover, the use of these elements provides a convenient place to indicate what kind of correspondence is intended as in the following retagging of the preceding example.

```
<title xml:id="shirley">Shirley</title>, which made
its Friday night debut only a month ago, was not
listed on <name xml:id="nbc">NBC</name>'s new schedule,
although <seg xml:id="network">the network</seg> says
<seg xml:id="show">the show</seg> still is being considered.

<linkGrp type="anaphoric_link" targFunc="antecedent anaphor">
  <link targets="#shirley #show"/>
  <link targets="#nbc #network"/>
</linkGrp>
```

In the following example, we may use exactly the same mechanism to express a correspondence amongst the anchors introduced following the fifth word after *English* in a text:

[9] The **corresp** attribute is thus distinct from the **target** attribute in that it is understood to create a double, rather than a single, link. It is also distinct from the **targets** attribute in that the latter lists all the identifiers of the elements that are doubly linked, whereas the **corresp** doubly links the element that bears the attribute with the element(s) that make up the value of the attribute.

```
English language. Except for not very<anchor xml:id="eng1"/>
<!-- ... -->
English at all at the time<anchor xml:id="eng2"/>
<!-- ... -->
English was still full of flaws<anchor xml:id="eng3"/>
<!-- ... -->
English. This was revised by young<anchor xml:id="eng4"/>
<!-- ... -->
<linkGrp type="five-word collocates">
<link type="collocates of ENGLISH" targets="#eng1 #eng2 #eng3 #eng4"/>
<!-- ... -->
</linkGrp>
```

16.4.2 Alignment of Parallel Texts

One very important application area for the alignment of parallel texts is multilingual corpora. Consider, for example, the need to align 'translation pairs' of sentences drawn from a corpus such as the Canadian Hansard, in which each sentence is given in both English and French. Concerning this problem, Gale and Church write:

> Most English sentences match exactly one French sentence, but it is possible for an English sentence to match two or more French sentences. The first two English sentences [in the example below] illustrate a particularly hard case where two English sentences align to two French sentences. No smaller alignments are possible because the clause '...sales...were higher...' in the first English sentence corresponds to (part of) the second French sentence. The next two alignments ... illustrate the more typical case where one English sentence aligns with exactly one French sentence. The final alignment matches two English sentences to a single French sentence. These alignments [which were produced by a computer program] agreed with the results produced by a human judge.[10]

The alignment produced by Gale and Church's program can be expressed in four different ways. The encoder must first decide whether to represent the alignment in terms of points within each text (using the <anchor> element) or in terms of whole stretches of text, using the <seg> element. To some extent the choice will depend on the process by which the software works out where alignment occurs, and the intention of the encoder. Secondly, the encoder may elect to represent the actual encoding using either **corresp** attributes attached to the individual <anchor> or <seg> elements, or using a free standing <linkGrp> element.

We present first a solution using <anchor> elements bearing only **corresp** attributes:

```
<div xml:lang="en" type="subsection">
 <p>
  <anchor corresp="#fa1" xml:id="ea1"/>According to our survey, 1988
   sales of mineral water and soft drinks were much higher than in 1987,
   reflecting the growing popularity of these products. Cola drink
   manufacturers in particular achieved above-average growth rates.
 <anchor corresp="#fa2" xml:id="ea2"/>The higher turnover was largely
   due to an increase in the sales volume.
 <anchor corresp="#fa3" xml:id="ea3"/>Employment and investment levels also climbed.
 <anchor corresp="#fa4" xml:id="ea4"/>Following a two-year transitional period,
   the new Foodstuffs Ordinance for Mineral Water came into effect on
   April 1, 1988. Specifically, it contains more stringent requirements
   regarding quality consistency and purity guarantees.</p>
```

[10] See Gale and Church (1993), from which the example in the text is taken.

```
</div>
<div xml:lang="fr" type="subsection">
  <p>
    <anchor corresp="#ea1" xml:id="fa1"/>Quant aux eaux minérales
     et aux limonades, elles rencontrent toujours plus d'adeptes. En effet,
     notre sondage fait ressortir des ventes nettement supérieures
     à celles de 1987, pour les boissons à base de cola
     notamment. <anchor corresp="#ea2" xml:id="fa2"/>La progression des
     chiffres d'affaires résulte en grande partie de l'accroissement
     du volume des ventes. <anchor corresp="#ea3" xml:id="fa3"/>L'emploi et
     les investissements ont également augmenté.
  <anchor corresp="#ea4" xml:id="fa4"/>La nouvelle ordonnance fédérale
     sur les denrées alimentaires concernant entre autres les eaux
     minérales, entrée en vigueur le 1er avril 1988 après
     une période transitoire de deux ans, exige surtout une plus
     grande constance dans la qualité et une garantie de la
     pureté.</p>
</div>
```

Source: [Gale and Church (1993)]

There is no requirement that the **corresp** attribute be specified in both English and French texts, since (as noted above) this attribute is defined as representing a mutual association. However, it may simplify processing to do so, and also avoids giving the impression that the English is translating the French, or vice versa. More seriously, this encoding does not make explicit that it is in fact the entire stretch of text between the anchors which is being aligned, not simply the points themselves. If for example one text contained material omitted from the other, this approach would not be appropriate.

We now present the same passage using the alternative <linkGrp> mechanism and marking explicitly the segments which have been aligned:

```
<div xml:id="div-e" xml:lang="en" type="subsection">
  <p>
    <seg xml:id="e_1">According to our survey, 1988 sales of mineral
        water and soft drinks were much higher than in 1987,
        reflecting the growing popularity of these products. Cola
        drink manufacturers in particular achieved above-average
        growth rates.</seg>
    <seg xml:id="e_2">The higher turnover was largely due to an
        increase in the sales volume.</seg>
    <seg xml:id="e_3">Employment and investment levels also climbed.</seg>
    <seg xml:id="e_4">Following a two-year transitional period, the new
        Foodstuffs Ordinance for Mineral Water came into effect on
        April 1, 1988. Specifically, it contains more stringent
        requirements regarding quality consistency and purity
        guarantees.</seg>
  </p>
</div>
<div xml:id="div-f" xml:lang="fr" type="subsection">
  <p>
    <seg xml:id="f_1">Quant aux eaux minérales et aux limonades,
        elles rencontrent toujours plus d'adeptes. En effet, notre
        sondage fait ressortir des ventes nettement
        supérieures à celles de 1987, pour les
        boissons à base de cola notamment.</seg>
    <seg xml:id="f_2">La progression des chiffres d'affaires
```

```
        résulte en grande partie de l'accroissement du volume
        des ventes.</seg>
    <seg  xml:id="f_3">L'emploi et les investissements ont
        également augmenté.</seg>
    <seg  xml:id="f_4">La nouvelle ordonnance fédérale sur
        les denrées alimentaires concernant entre autres les
        eaux minérales, entrée en vigueur le 1er avril
        1988 après une période transitoire de deux
        ans, exige surtout une plus grande constance dans la
        qualité et une garantie de la pureté.</seg>
  </p>
</div>
<linkGrp  type="alignment"  domains="div-e div-f">
  <link  targets="#e_1 #f_1"/>
  <link  targets="#e_2 #f_2"/>
  <link  targets="#e_3 #f_3"/>
  <link  targets="#e_4 #f_4"/>
</linkGrp>
```

Note that use of the <ab> element allows us to mark up the orthographic sentences in both languages independently of the alignment: the first translation pair in this example might be marked up as follows:

```
<div  xml:id="english"  xml:lang="en"  type="subsection">
  <ab  xml:id="english1">
    <s>According to our survey, 1988 sales of mineral water and soft
        drinks were much higher than in 1987, reflecting the growing popularity
        of these products.</s>
    <s>Cola drink manufacturers in particular achieved above-average
        growth rates.</s>
  </ab>
</div>
<div  xml:id="french"  xml:lang="fr"  type="subsection">
  <ab  xml:id="french1">
    <s  xml:id="fs1">Quant aux eaux minérales et aux limonades, elles
        rencontrent toujours plus d'adeptes.</s>
    <s  xml:id="fs2">En effet, notre sondage fait ressortir des ventes nettement
        supérieures à celles de 1987, pour les boissons à
        base de cola notamment.</s>
  </ab>
</div>
```

16.4.3 A Three-way Alignment

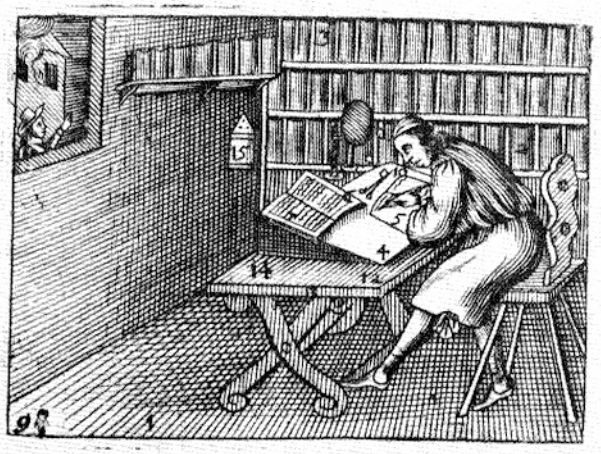
The preceding encoding of the alignment of parallel passages from two texts requires that those texts and the alignment all be part of the same document. If the texts are in separate documents, then complete URIs, whether absolute or relative (section *16. Linking, Segmentation, and Alignment*), will be required. These external pointers may appear anywhere within the document, but if they are created solely for use in encoding links, they may for convenience be grouped within the <linkGrp> (or other grouping element that uses them for linking).

To demonstrate this facility, we consider how we might encode the alignments in an extract from Comenius' *Orbis Sensualium Pictus*, in the English translation of Charles Hoole (1659).

Each topic covered in this work has three parts: a picture, a prose text in Latin describing the topic, and a carefully-aligned translation of the Latin into English, German, or some other vernacular. Key terms in the

XCVIII.

The Study. *Muſéum.*



The Study 1.
to a place
where a Student, 2.
a part from men,
ſitteth alone,
addicted to his Studies,
whilſt he readeth
Books, 3.
which being within
his reach, he layeth
open upon a Desk 4.
and picketh all the
beſt things out of
them into his own
Manual, 5.

Muſéum 1.
eſt locus,
ubi *Studioſus*, 2.
ſecretus ab hominibus,
ſolus ſedet,
Studiis deditus,
dum lectitat
Libros, 3.
quos penes ſe
ſuper *Pluteum* 4.
exponit, & ex iliis
in *Manuale* 5. ſuum
optima quæq; excerpit,

two texts are typographically distinct, and are linked to the picture by numbers, which appear in the two texts and within the picture as well.

First, we consider the text portions. The English and Latin portions have been encoded as distinct elements. Identifiers have been attached to each typographic line, but no other encoding added, to simplify the example.

```
<div xml:id="e98" xml:lang="en" type="lesson">
  <head>The Study</head>
  <p>
    <seg xml:id="e9801">The Study</seg>
    <seg xml:id="e9802">is a place</seg>
    <seg xml:id="e9803">where a Student,</seg>
    <seg xml:id="e9804">a part from men,</seg>
    <seg xml:id="e9805">sitteth alone,</seg>
    <seg xml:id="e9806">addicted to his Studies,</seg>
    <seg xml:id="e9807">whilst he readeth</seg>
    <seg xml:id="e9808">Books,</seg>
  </p>
</div>
<div xml:id="l98" xml:lang="la" type="lesson">
  <head>Muséum</head>
  <p>
    <seg xml:id="l9801">Museum</seg>
    <seg xml:id="l9802">est locus</seg>
    <seg xml:id="l9803">ubi Studiosus,</seg>
    <seg xml:id="l9804">secretus ab hominibus,</seg>
    <seg xml:id="l9805">solus sedet,</seg>
    <seg xml:id="l9806">Studiis deditus,</seg>
    <seg xml:id="l9807">dum lectitat</seg>
    <seg xml:id="l9808">Libros,</seg>
  </p>
</div>
```

Source: [41]

Next we consider the non-textual parts of the page. Encoding this requires providing two distinct components: firstly a digitized rendering of the page itself, and secondly a representation of the areas within that image which are to be aligned. In section *11.1. Digital Facsimiles* we present a simple way of doing this using the TEI-defined markup for alignment of facsimiles. In the present chapter we demonstrate a more powerful means of aligning arbitrary polygons and points, which uses the XML notation SVG (Scalable Vector Graphics). This provides appropriate facilities for both these requirements:

```
 <svg>
<image
xlink:href="p1764.png"
width="597" height="897"
id="p981" />
<rect id="p982" x="75" y="75" width="25" height="10"/>
<rect id="p983" x="55" y="42" width="25" height="10"/>
</svg>
```

This example of SVG defines two rectangles at the locations with the specified x and y coordinates. A view is defined on these, enabling them to be mapped by an SVG processor to the image found at the URL specified (p1764.png). It also defines unique identifiers for the whole image, and the two views of it, which we will use

within our alignment, as shown next (for further discussion of the handling of images and graphics, see section *14.3. Specific Elements for Graphic Images*; for further discussion of using non-TEI XML vocabularies such as SVG within a TEI document, see section *22.6. Combining TEI and Non-TEI Modules*).

As printed, the Comenius text exhibits three kinds of alignment.

1. The English and Latin portions are printed in two parallel columns, with corresponding phrases, (represented above by <seg> elements), more or less next to each other.

2. Particular words or phrases are marked as terms in the two languages by a change of rendition: the English text, which otherwise uses black letter type throughout, has the words *The Study*, *a Student*, *Studies*, and *Books* in a roman font; in the Latin text, which is printed in roman, the corresponding words (*Museum*, *Studiosus*, *Studiis*, and *Libros*) are all in italic.

3. Numbered labels appear within the text portions, linking keywords to each other and to sections of the picture. These labels, which have been left out of the above encoding, are attached to the first, third, and last segments in each language quoted below, and also appear (rather indistinctly) within the picture itself. Thus, the images of the study, the student, and his books are each aligned with the correct term for them in the two languages.

The first kind of alignment might be represented by using the **corresp** attribute on the <seg> element. The second kind might be represented by using the <gloss> and <term> mechanism described in section *3.3.4. Terms, Glosses, Equivalents, and Descriptions*. The third kind of alignment might be represented using pointers embedded within the texts, for example:

```
    ...
    <seg xml:id="e9803">where a <ref n="2" target="#p982">Student</ref>,</seg>
    <seg xml:id="l9803">ubi <ref n="2" target="#p982">Studiosus</ref>,</seg>
    ...
```

We choose however to use the <link> element, since this provides a more efficient way of representing the three-way alignment between English, Latin, and picture without redundancy.

```
<linkGrp type="alignment">
  <link targets="#e9801 #l9801 #p981"/>
  <link targets="#e9802 #l9802"/>
  <link targets="#e9803 #l9803 #p982"/>
  <link targets="#e9804 #l9804"/>
  <link targets="#e9805 #l9805"/>
  <link targets="#e9806 #l9806"/>
  <link targets="#e9807 #l9807"/>
  <link targets="#e9808 #l9808 #p983"/>
</linkGrp>
```

This map, of course, only aligns whole segments and image portions, since these are the only parts of our encoding which bear identifiers and can therefore be pointed to. To add to it the alignment between the typographically distinct words mentioned above, new elements must be defined, either within the text itself or externally by using stand off techniques. Encoding these word pairs as <term> and <gloss>, although intuitively obvious, requires a non-trivial decision as to whether the Latin text is glossing the English, or vice-versa. Tagging all the marked words as <term> avoids the difficult decision, but might be thought by some encoders to convey the wrong information about the words in question. Simply tagging them as additional embedded <seg> elements with identifiers that can be aligned like the others is also a possibility.

These solutions all require the addition of further markup to the text. This may pose no problems, or it may be infeasible, for example because the text is held on a read-only medium. If it is not feasible to add more markup to the original text, some form of stand-off markup will be needed. Any item within the text that can be pointed to using the various pointer schemes discussed in this chapter may be used, not simply those which rely on the existence of an **xml:id** attribute. For example, if the segments in our example did not have identifiers, they could still be addressed using the notation introduced in *16.2.3. W3C element() Scheme* above. Suppose our example had been more lightly tagged, as follows:

```
<div xml:id="E98" xml:lang="en" type="lesson">
  <head>The Study</head>
  <ab>The Study</ab>
  <ab>is a place</ab>
  <ab>where a Student,</ab>
</div>
<div xml:id="L98" xml:lang="la" type="lesson">
  <head>Muséum</head>
  <ab>Museum</ab>
  <ab>est locus</ab>
  <ab>ubi Studiosus,</ab>
</div>
```

Source: [41]

To express the same alignment mentioned above, we could use an XPath expression to identify the required <seg> elements:

```
<linkGrp type="alignment">
  <link
     targets="#element(L98/2) #element(E98/2) #p981"/>
  <link targets="#element(L98/3) #element(E98/3)"/>
</linkGrp>
```

In the absence of any markup around individual substrings of the element content, the string-range pointer scheme discussed in *16.2.4.5. string-range(pointer, offset [, length])* may also be helpful: for example, to indicate that the words *Studies* and *Studiis* correspond, we might express the link between them as follows:

```
<link
   targets="#string-range(xpath1(id('e9806')),16,7) #string-range(xpath1(id('l9806')),0,7)"/>
```

16.5 Synchronization

In the previous section we discussed two particular kinds of alignment: alignment of parallel texts in different languages; and alignment of texts and portions of an image. In this section we address another specialized form of alignment: synchronization. The need to mark the relative positions of text components with respect to time arises most naturally and frequently in transcribed spoken texts, but it may arise in any text in which quoted speech occurs, or events are described within a time frame. The methods described here are also generalizable for other kinds of alignment (for example, alignment of text elements with respect to space).

16.5.1 Aligning Synchronous Events

Provided that explicit elements are available to represent the parts or places to be synchronized, then the global linking attribute **synch** may be used to encode such synchronization, once it has been identified.

att.global.linking defines a set of attributes for hypertext and other linking, which are enabled for all elements when the additional tag set for linking is selected.

@**synch** (synchronous) points to elements that are synchronous with the current element.

This is another of the attributes made globally available by the mechanism described in the introduction to this chapter. Alternatively, the <link> and <linkGrp> elements may be used to make explicit the fact that the synchronous elements are aligned.

To illustrate the use of these mechanisms for marking synchrony, consider the following representation of a spoken text:

```
B: The first time in twenty five years, we've cooked Christmas
(unclear) for a blooming great load of people.
A: So you're [1] (unclear) [2]
B: [1] It will be [2] nice in a way, but, [3] be strange. [4]
A: [3] Yeah [4], yeah, cos it, it's [5] the [6]
B: [5] not [6]
```

This representation uses numbers in brackets to mark the points at which speakers overlap each other. For example, the *[1]* in A's first speech is to be understood as coinciding with the *[1]* in B's second speech.[11]

To encode this we use the spoken texts module, described in chapter *8. Transcriptions of Speech*, together with the module described in the present chapter. First, we transcribe this text, marking the synchronous points with <anchor> elements, and providing a **synch** attribute on one of each of the pairs of synchronous anchors. As noted in the example given above (section *16.4.2. Alignment of Parallel Texts*), correspondence, and hence synchrony, is a symmetric relation; therefore the attribute need only be specified on one of the pairs of synchronous anchors.

```
<div xml:id="BNC-d1" type="convers">
 <u xml:id="u2b" who="#b"> The first time in twenty five years,
    we've cooked Christmas <unclear> for a blooming great
       load of people.</unclear>
 </u>
 <u xml:id="u3a" who="#a">So you're
 <anchor synch="#t1b" xml:id="t1a"/>
   <unclear>
    <anchor synch="#t2b" xml:id="t2a"/>
   </unclear>
 </u>
 <u xml:id="u3b" who="#b">
   <anchor xml:id="t1b"/>It will be <anchor xml:id="t2b"/>
    nice in a way, but, <anchor xml:id="t3b"/>
    be strange.<anchor xml:id="t4b"/>
 </u>
 <u xml:id="u4a" who="#a">
   <anchor synch="#t3b" xml:id="t3a"/>Yeah
 <anchor synch="#t4b" xml:id="t4a"/>, yeah, cos it, its
 <anchor synch="#t5b" xml:id="t5a"/>the
 <anchor synch="#t6b" xml:id="t6a"/>
 </u>
 <u xml:id="u4b" who="#b">
   <anchor xml:id="t5b"/>not<anchor xml:id="t6b"/>
 </u>
```

[11] This sample is taken from a conversation collected and transcribed for the British National Corpus.

```
<!-- ... -->
</div>
```

Source: [20]

We can encode this same example using <link> and <linkGrp> elements to make the temporal alignment explicit. A <back> element has been used to enclose the <linkGrp> element, but the links may be located anywhere the encoder finds convenient:

```
<back>
  <linkGrp
     xml:id="lg1"
     domains="BNC-d1 BNC-d1"
     targFunc="speaker.a speaker.b"
     type="synchronous_alignment">
    <link xml:id="l1" targets="#t1a #t1b"/>
    <link xml:id="l2" targets="#t2a #t2b"/>
    <link xml:id="l3" targets="#t3a #t3b"/>
    <link xml:id="l4" targets="#t4a #t4b"/>
    <link xml:id="l5" targets="#t5a #t5b"/>
    <link xml:id="l6" targets="#t6a #t6b"/>
  </linkGrp>
</back>
```

The **xml:id** attributes are provided for the <link> and <linkGrp> elements here for reasons discussed in the next section, *16.5.2. Placing Synchronous Events in Time*.

As with other forms of alignment, synchronization may be expressed between stretches of speech as well as between points. When complete utterances are synchronous, for example, if one person says *What?* and another *No!* at the same time, that can be represented without <anchor> elements as follows.

```
<u synch="#u02" xml:id="u01" who="#a">What?</u>
<u xml:id="u02" who="#b">No!</u>
```

A simple way of expressing overlap (where one speaker starts speaking before another has finished) is thus to use the <seg> element to encode the overlapping portions of speech. For example,

```
<u who="#a"> So you're <unclear synch="#u-b1"/>
</u>
<u who="#b">
  <seg xml:id="u-b1"> It will be </seg> nice in a way, but,
<seg synch="#u-a3"> be strange. </seg>
</u>
<u who="#a">
  <seg xml:id="u-a3"> Yeah </seg>, yeah, cos it,
its <seg synch="#u-b2"> the </seg>
</u>
<u xml:id="u-b2" who="#b"> not </u>
```

Note in this encoding how synchronization has been effected between an empty <unclear> element and the content of a <seg> element, and between the content of an <u> element and that of another <seg>, using the **synch** attribute. Alternatively, a <linkGrp> could be used in the same way as above.

16.5.2 Placing Synchronous Events in Time

A synchronous alignment specifies which points in a spoken text occur at the same time, and the order in which they occur, but does not say at what time those points actually occur. If that information is available to the encoder it can be represented by means of the <when> and <timeline> elements, whose description and attributes are the following:

<when/> indicates a point in time either relative to other elements in the same timeline tag, or absolutely.

@**absolute** supplies an absolute value for the time.

@**interval** specifies the numeric portion of a time interval

@**unit** specifies the unit of time in which the **interval** value is expressed, if this is not inherited from the parent <timeline>.

@**since** identifies the reference point for determining the time of the current <when> element, which is obtained by adding the interval to the time of the reference point.

<timeline> (timeline) provides a set of ordered points in time which can be linked to elements of a spoken text to create a temporal alignment of that text.

@**origin** designates the origin of the timeline, i.e. the time at which it begins.

@**interval** specifies the numeric portion of a time interval

@**unit** specifies the unit of time corresponding to the **interval** value of the timeline or of its constituent points in time.

Each <when> element indicates a point in time, either directly by means of the **absolute** attribute, whose value is a string which specifies a particular time, or indirectly by means of the **since** attribute, which points to another <when>. If the **since** is used, then the **interval** and **unit** attributes should also be used to indicate the amount of time that has elapsed since the time specified by the element pointed to by the **since** attribute; the value **-1** can be given to indicate that the interval is unknown.

If the <when> elements are uniformly spaced in time, then the **interval** and **unit** values need be given once in the <timeline>, and not repeated in any of the <when> elements. If the intervals vary, but the units are all the same, then the **unit** attribute alone can be given in the <timeline> element, and the **interval** attribute given in the <when> element.

The **origin** attribute in the <timeline> element points to a <when> element which specifies the reference or origin for the timings within the <timeline>; this must, of course, specify its position in time absolutely.

The following <timeline> might be used to accompany the marked up conversation shown in the preceding section:

```
<timeline xml:id="tl1" origin="#w0" unit="ms">
  <when xml:id="w0" absolute="11:30:00"/>
  <when xml:id="w1" interval="unknown" since="#w0"/>
  <when xml:id="w2" interval="100" since="#w1"/>
  <when xml:id="w3" interval="200" since="#w2"/>
  <when xml:id="w4" interval="150" since="#w3"/>
  <when xml:id="w5" interval="250" since="#w4"/>
  <when xml:id="w6" interval="100" since="#w5"/>
</timeline>
```

The information in this <timeline> could now be linked to the information in the <linkGrp> which provides the temporal alignment (synchronization) for the text, as follows:

```
<linkGrp
   type="temporal_specification"
   domains="lg1 tl1"
   targFunc="synch.points when">
  <link  targets="#l1 #w1"/>
  <link  targets="#l2 #w2"/>
  <link  targets="#l3 #w3"/>
  <link  targets="#l4 #w4"/>
  <link  targets="#l5 #w5"/>
  <link  targets="#l6 #w6"/>
</linkGrp>
```

To avoid the need for two distinct link groups (one marking the synchronization of anchors with each other, and the other marking their alignment with points on the time line) it would be better to link the <when> elements with the synchronous points directly:

```
<linkGrp
   type="temporal_specification"
   domains="BNC-d1 BNC-d1 tl1"
   targFunc="speaker.a speaker.b when">
  <link  targets="#t1a #t1b #w1"/>
  <link  targets="#t2a #t2b #w2"/>
  <link  targets="#t3a #t3b #w3"/>
  <link  targets="#t4a #t4b #w4"/>
  <link  targets="#t5a #t5b #w5"/>
  <link  targets="#t6a #t6b #w6"/>
</linkGrp>
```

Finally, suppose that a digitized audio recording is also available, and an XML file that assigns identifiers to the various temporal spans of sound is available. For example, the following Synchronized Multimedia Integration Language (SMIL, pronounced "smile") fragment:

```
    <audio src="rtsp://soundstage.pi.cnr.it:554/home/az/bncSound/xmas4lots.mp3"
    xml:id="au1" begin="05.2s" />
    <audio src="rtsp://soundstage.pi.cnr.it:554/home/az/bncSound/xmas4lots.mp3"
    xml:id="au2" begin="05.7s" />
    <audio src="rtsp://soundstage.pi.cnr.it:554/home/az/bncSound/xmas4lots.mp3"
    xml:id="au3" begin="05.9s" />
    <audio src="rtsp://soundstage.pi.cnr.it:554/home/az/bncSound/xmas4lots.mp3"
    xml:id="au4" begin="06.3s" />
    <audio src="rtsp://soundstage.pi.cnr.it:554/home/az/bncSound/xmas4lots.mp3"
    xml:id="au5" begin="06.9s" />
    <audio src="rtsp://soundstage.pi.cnr.it:554/home/az/bncSound/xmas4lots.mp3"
    xml:id="au6" begin="07.4s" />
```

URIs pointing to the <audio> elements could also be included as a fourth component in each of the above <link> elements, thus providing a synchronized audio track to complement the transcribed text.

For further discussion of this and related aspects of encoding transcribed speech, refer to chapter *8. Transcriptions of Speech*.

16.6 Identical Elements and Virtual Copies

This section introduces the notion of a virtual element, that is, an element which is not explicitly present in a text, but the presence of which an application can infer from the encoding supplied. In this section, we

are concerned with virtual elements made by simply cloning existing elements. In the next section (*16.7. Aggregation*), we discuss virtual elements made by aggregating existing elements.

Provided that explicit elements are available to represent the parts or places to be linked, then the global linking attributes **sameAs** and **copyOf** may be used to encode this kind of equivalence:

att.global.linking defines a set of attributes for hypertext and other linking, which are enabled for all elements when the additional tag set for linking is selected.

@**sameAs** points to an element that is the same as the current element.

@**copyOf** points to an element of which the current element is a copy.

It is useful to be able to represent the fact that one element of text is identical to others, for analytical purposes, or (especially if the elements have lengthy content) to obviate the need to repeat the content. For example, consider the repetition of the <date> element in the following material:

```
<p>In small clumsy letters he wrote:
<q  rend="centered italic">
   <date  xml:id="d840404">April 4th,
       1984</date>.</q>
</p>
<p>He sat back. A sense of complete helplessness had
descended upon him. ...</p>
<p>His small but childish handwriting straggled up
and down the page, shedding first its capital letters
and finally even its full stops:
<q  rend="italic">
   <date>April 4th, 1984</date>.
    Last night to the flicks. ... </q>
</p>
```

Source: [150]

Suppose now that we wish to encode the fact that the second <date> element above has identical content to the first. The **sameAs** attribute is provided for this purpose. Using it, we could recode the last line of the above example as follows:

```
<date  sameAs="#d840404">April 4th,
1984</date>
Last night to the flicks ...
```

The **sameAs** attribute may be used to document the fact that two elements have identical content. It may be regarded as a special kind of link. It should only be attached to an element with identical content to that which it targets, or to one the content of which clearly designates it as a repetition, such as the word *repeat* or *bis* in the representation of the chorus of a song, the second time it is to be sung. The relation specified by the **sameAs** attribute is symmetric: if a chorus is repeated three times and each repetition bears a **sameAs** attribute indicating the first occurrence of the element concerned, it is implied that each chorus is identical, and there is no need for the first occurrence to specify any of its copies.

The **copyOf** attribute is used in a similar way to indicate that the content of the element bearing it is identical to that of another. The difference is that the content is not itself repeated. The effect of this attribute is thus to create a virtual copy of the element indicated. Using this attribute, the repeated date in the first example above could be recoded as follows:

```
<date rend="italic" copyOf="#d840404"/>
```

An application program should replace whatever is the actual content of an element bearing a **copyOf** attribute with the content of the element specified by it. If the content of the element specified includes other elements, these will become embedded within the element bearing the attribute. Care must be taken to ensure that the document is valid both before and after this embedding takes place. If, for example, the element bearing a **copyOf** attribute requires a mandatory sub-component, then this component must be present (though possibly empty), even though it will be replaced by the content of the targetted element.

The following example demonstrates how the **copyOf** attribute may be used in conjunction with the <seg> element to highlight the differences between almost identical repetitions:

```
<sp>
  <speaker>Mikado</speaker>
  <l>My <seg xml:id="Mik-l1s">object all sublime</seg>
  </l>
  <l>I shall <seg xml:id="Mik-l2s">achieve in time</seg>—</l>
  <l xml:id="Mik-l3">To let <seg xml:id="l3s">the punishment fit the crime</seg>,</l>
  <l xml:id="Mik-l4">
    <seg copyOf="#Mik-l3s"/>;</l>
  <l xml:id="Mik-l5">And make each pris'ner pent</l>
  <l xml:id="Mik-l6">Unwillingly represent</l>
  <l xml:id="Mik-l7">A source <seg xml:id="Mik-l7s">of innocent merriment</seg>,</l>
  <l xml:id="Mik-l8">
    <seg copyOf="#Mik-l7s"/>!</l>
</sp>
<sp>
  <speaker>Chorus</speaker>
  <l>His <seg copyOf="#Mik-l1s"/>
  </l>
  <l>He will <seg copyOf="#Mik-l2s"/>
  </l>
  <l copyOf="#Mik-l3"/>
  <l copyOf="#Mik-l4"/>
  <l copyOf="#Mik-l5"/>
  <l copyOf="#Mik-l6"/>
  <l copyOf="#Mik-l7"/>
  <l copyOf="#Mik-l8"/>
</sp>
```

Source: [86]

For further examples of the use of this attribute, see *16.8. Alternation* and *19.3. Another Tree Notation*.

16.7 Aggregation

Because of the strict hierarchical organization of elements, or for other reasons, it may not always be possible or desirable to include all the parts of a possibly fragmented text segment within a single element. In section *16.1.4. Intermediate Pointers* we introduced the notion of an intermediate pointer as a way of pointing to discontinuous segments of this kind. In this section we first describe another way of linking the parts of a discontinuous whole, using a set of linking attributes, which are made available for any tag by following the procedure described at the beginning of this chapter. We then describe how the <link> element may be used to aggregate such segments, and finally introduce the <join> element, which is a special-purpose linking element specifically for representing the aggregation of parts, and the <joinGrp> for grouping <join> elements.

The linking attributes for aggregation are **next** and **prev**; each of these attributes has a single identifier as its value:

att.global.linking defines a set of attributes for hypertext and other linking, which are enabled for all elements when the additional tag set for linking is selected.

@**next** points to the next element of a virtual aggregate of which the current element is part.

@**prev** (previous) points to the previous element of a virtual aggregate of which the current element is part.

The <join> element is also a member of the class of att.pointing elements, and so may carry any of the attributes of that class; for the list, see section *16.1. Links*.

Here is the material on which we base our first illustration of the use of these mechanisms. Our problem is to represent the s-units identified below as **qs3** and **qs4** as a single (but discontinuous) whole:

```
<q>
  <s  xml:id="qs2">Monsieur Paul, after he has taken equal
     parts of goose breast and the finest pork, and
     broken a certain number of egg yolks into them,
     and ground them <emph>very</emph>, very fine,
     cooks all with seasoning for some three hours.</s>
  <s  xml:id="qs3">
    <emph>But</emph>,</s>
</q>
<s  xml:id="ps2">she pushed her face nearer, and looked with
ferocious gloating at the pâté
inside me, her eyes like X rays,</s>
<q>
  <s  xml:id="qs4">he never stops stirring it!</s>
  <s  xml:id="qs5">Figure to yourself the work of it —</s>
  <s  xml:id="qs6">stir, stir, never stopping!</s>
</q>
```

Source: [74]

Using the **prev** and **next** attributes, we can link the s-units with identifiers **qs3** and **qs4**, either singly or doubly as follows:

```
    <s xml:id="qs3" next="#qs4"><emph>But</emph>,</s>
    <s xml:id="qs4">he never stops stirring it!</s>
```

```
    <s xml:id="qs3"><emph>But</emph>,</s>
    <s xml:id="qs4" prev="#qs3">he never stops stirring it!</s>
```

```
    <s xml:id="qs3" next="#qs4"><emph>But</emph>,</s>
    <s xml:id="qs4" prev="#qs3">he never stops stirring it!</s>
```

Double linking of the two s-units, as illustrated by the last of these encodings, is equivalent to specifying a <link> element:

```
<link  type="join"  targets="#qs3 #qs4"/>
```

Such a <link> element must carry a **type** attribute with a value of **join** to specify that the link is to be understood as joining its targets into a single aggregate.

The <join> element is equivalent to a <link> element of type **join**.

Unlike the <link> element, the <join> element can additionally specify information about the virtual element which it represents, by means of its **result** attribute. And finally, unlike the <link> element, the position of a <join> element within a text is significant: it must be supplied at a position where the element indicated by its **result** attribute would be contextually legal.

<join> identifies a possibly fragmented segment of text, by pointing at the possibly discontiguous elements which compose it.

@**result** specifies the name of an element which this aggregation may be understood to represent.

@**targets** specifies the identifiers of the elements or passages to be joined into a virtual element.

<joinGrp> (join group) groups a collection of join elements and possibly pointers.

@**result** describes the result of the joins gathered in this collection.

To conclude the above example, we now use a <join> element to represent the virtual sentence formed by the aggregation of **s1** and **s2**:

```
<join  targets="#qs3 #qs4"  result="s"/>
```

As a further example, consider the following list of authors' names. The object of the <join> element here is to provide another list, composed of those authors from the larger list who happen to come from Heidelberg:

```
<list>
  <head>Authors</head>
  <item  xml:id="a_uf">Figge, Udo </item>
  <item  xml:id="a_ch">Heibach, Christiane </item>
  <item  xml:id="a_gh">Heyer, Gerhard </item>
  <item  xml:id="a_bp">Philipp, Bettina </item>
  <item  xml:id="a_ms">Samiec, Monika </item>
  <item  xml:id="a_ss">Schierholz, Stefan </item>
</list>
<join  targets="#a_ch #a_bp #a_ss"  result="list">
  <desc>Authors from Heidelberg</desc>
</join>
```

The following example shows how <join> can be used to reconstruct a text cited in fragments presented out of order. The poem being remembered (an unusual translation of a well-known poem by Basho) runs 'When the old pond / gets a new frog, / it's a new pond.'

```
<sp>
  <speaker>Hughie</speaker>
  <p>How does it go?
  <q>
     <l  xml:id="frog-x1">da-da-da</l>
```

```
     <l  xml:id="frog-l2">gets a new frog</l>
     <l>...</l>
    </q>
  </p>
</sp>
<sp>
  <speaker>Louie</speaker>
  <p>
    <q>
     <l  xml:id="frog-l1">When the old pond</l>
     <l>...</l>
    </q>
  </p>
</sp>
<sp>
  <speaker>Dewey</speaker>
  <p>
    <q>...
    <l  xml:id="frog-l3">It's a new pond.</l>
    </q>
  </p>
  <join  targets="#frog-l1 #frog-l2 #frog-l3"  result="lg"  scope="root"/>
</sp>
```

As with other forms of link, a grouping element <joinGrp> is available for use when a number of <join> elements of the same kind co-occur. This avoids the need to specify the **result** attribute for each <join> if they are all of the same type, and also allows us to restrict the domain within which their target elements are to be found, in the same way as for <linkGrp> elements (see *16.1.3. Groups of Links*). Like a <join>, a <joinGrp> may appear only where the elements represented by its contents are legal. Thus if we had created many <join> tags of the sort just described, we could group them together, and require that their components are all contained by an element with the identifier **MFKFhungry** as follows:

```
<joinGrp  domains="mfkfhungry mfkfhungry"  result="s">
  <join  targets="#qs3 #qs4"/>
  <join  targets="#qs5 #qs6"/>
</joinGrp>
```

The <join> element is useful as a means of representing non-hierarchic structures (as further discussed in chapter *20. Non-hierarchical Structures*). It may also be used as a convenient way of representing a variety of analytic units, like the and <interp> elements discussed in chapter *17. Simple Analytic Mechanisms.* As an example, consider the following famous Zen koan:

> Zui-Gan called out to himself every day, 'Master.'
>
> Then he answered himself, 'Yes, sir.'
>
> And then he added, 'Become sober.'
>
> Again he answered, 'Yes, sir.'
>
> 'And after that,' he continued, 'do not be deceived by others.'
>
> 'Yes, sir; yes, sir,' he replied.

Suppose now that we wish to represent an interpretation of the above passage in which we distinguish between the various 'voices' adopted by Zui-Gan. In the following encoding, the **who** attribute has been used

for this purpose; its value on each occasion supplies a pointer to the 'voice' to which each speech is attributed. (For convenience in this example, we use simply the first occurrence of the names used for each voice as the target for these pointers.) Note also that we add **xml:id** attributes to each distinct speech fragment, which we can then use to link the material spoken by each voice:

```
<text xml:id="zuitxt">
 <body>
  <p>
   <name xml:id="zuigan">Zui-Gan</name> called out to himself every day,
  <q next="#zuiq2" xml:id="zuiq1" who="#zuigan">
     <name xml:id="master">Master</name>.</q>
  </p>
  <p>Then he answered himself,
  <q next="#zuiq4" xml:id="zuiq2" who="#zuigan">Yes, sir.</q>
  </p>
  <p>And then he added,
  <q next="#zuiq5" xml:id="zuiq3" who="#master">Become sober.</q>
  </p>
  <p>Again he answered,
  <q next="#zuiq7" xml:id="zuiq4" who="#zuigan">Yes, sir.</q>
  </p>
  <p>
   <q next="#zuiq6" xml:id="zuiq5" who="#master">And after that,</q>
     he continued,
  <q xml:id="zuiq6" who="#master">do not be deceived by others.</q>
  </p>
  <p>
   <q xml:id="zuiq7" who="#zuigan">Yes, sir; yes, sir,</q>
     he replied.</p>
 </body>
</text>
```

Source: [213]

However, by using the <join> element, we can directly represent the complete speech attributed to each voice:

```
<joinGrp result="q">
 <join targets="#zuiq1 #zuiq2 #zuiq4 #zuiq7">
  <desc>what Zui-Gan said</desc>
 </join>
 <join targets="#zuiq3 #zuiq5 #zuiq6">
  <desc>what Master said</desc>
 </join>
</joinGrp>
```

Note the use of the <desc> child element within the two <join>s making up the <q> element here. These enable us to document the speakers of the two virtual <q> elements represented by the <join> elements; this is necessary because the there is no way of specifying the attributes to be associated with a virtual element, in particular there is no way to specify a **who** value for them.

Suppose now that **xml:id** attributes, for whatever reasons, are not available. Then <ptr> elements may be created using any of the methods described in sections *16.2.3. W3C element() Scheme* or *16.2.4. TEI XPointer Schemes*. The **xml:id** attributes of *these* elements may now be specified by the **targets** attribute on the <join> elements.

```
<text>
  <body>
<!-- five div1 elements -->
    <div1>
      <p>Zui-Gan called out to himself every day, <q>Master.</q>
      </p>
      <p>Then he answered himself, <q>Yes, sir.</q>
      </p>
      <p>And then he added, <q>Become sober.</q>
      </p>
      <p>Again he answered, <q>Yes, sir.</q>
      </p>
      <p>
        <q>And after that,</q> he continued, <q>do not be deceived by others.</q>
      </p>
      <p>
        <q>Yes, sir; yes, sir,</q> he replied.</p>
      <ab  type="aggregation">
        <ptr  xml:id="rzuiq1"  target="./#xpath1(//div1[6]/p[1]/q[1])"/>
        <ptr  xml:id="rzuiq2"  target="./#xpath1(//div1[6]/p[2]/q[1])"/>
        <ptr  xml:id="rzuiq3"  target="./#xpath1(//div1[6]/p[3]/q[1])"/>
        <ptr  xml:id="rzuiq4"  target="./#xpath1(//div1[6]/p[4]/q[1])"/>
        <ptr  xml:id="rzuiq5"  target="./#xpath1(//div1[6]/p[5]/q[1])"/>
        <ptr  xml:id="rzuiq6"  target="./#xpath1(//div1[6]/p[5]/q[2])"/>
        <ptr  xml:id="rzuiq7"  target="./#xpath1(//div1[6]/p[6]/q[1])"/>
        <joinGrp  evaluate="one"  result="q">
          <join  targets="#rzuiq1 #rzuiq2 #rzuiq4 #rzuiq7">
            <desc>what Zui-Gan said</desc>
          </join>
          <join  targets="#rzuiq3 #rzuiq5 #rzuiq6">
            <desc>what Master said</desc>
          </join>
        </joinGrp>
      </ab>
    </div1>
  </body>
</text>
```

Source: [213]

The extended pointer with identifier **rzuiq2**, for example, may be read as 'the first <q> in the first <p>, within the sixth <div1> element of the current document.'

16.8 Alternation

This section proposes elements for the representation of alternation. We say that two or more elements are in exclusive alternation if any of those elements could be present in a text, but one and only one of them is; in addition, we say that those elements are mutually exclusive. We say that the elements are in inclusive alternation if at least one (and possibly more) of them is present. The elements that are in alternation may also be called alternants.

The need to mark exclusive alternation arises frequently in text encoding. A common situation is one in which it can be determined that exactly one of several different words appears in a given location, but it cannot be determined which one. One way to mark such an exclusive alternation is to use the linking attribute **exclude**. Having marked an exclusive alternation, it can sometimes later be determined which of the alternants actually appears in the given location. To preserve the fact that an alternation was posited, one can add the linking attribute **select** to a tag which hierarchically encompasses the alternants, which points to the one which

actually appears. To assign responsibility and degree of certainty to the choice, one can use the <certainty> tag described in chapter *21. Certainty and Responsibility*. Also see that chapter for further discussion of certainty in general.

The **exclude** and **select** attributes may be used with any element assuming that they have been declared following the procedure discussed in the introduction to this chapter.

att.global.linking defines a set of attributes for hypertext and other linking, which are enabled for all elements when the additional tag set for linking is selected.

@**exclude** points to elements that are in exclusive alternation with the current element.

@**select** selects one or more alternants; if one alternant is selected, the ambiguity or uncertainty is marked as resolved. If more than one alternant is selected, the degree of ambiguity or uncertainty is marked as reduced by the number of alternants not selected.

A more general way to mark alternation, encompassing both exclusive and inclusive alternation, is to use the linking element <alt>. The description and attributes of this tag and of the associated grouping tag <altGrp> are as follows. These elements are also members of the att.pointing class and therefore have all the attributes associated with that class.

<alt/> (alternation) identifies an alternation or a set of choices among elements or passages.

@**targets** specifies the identifiers of the alternative elements or passages.

@**weights** If **mode** is `excl`, each weight states the probability that the corresponding alternative occurs. If **mode** is **incl** each weight states the probability that the corresponding alternative occurs given that at least one of the other alternatives occurs.

<altGrp> (alternation group) groups a collection of <alt> elements and possibly pointers.

To take a simple hypothetical example, suppose in transcribing a spoken text, we encounter an utterance that we can understand either as *We had fun at the beach today.* or as *We had sun at the beach today.* We can represent the exclusive alternation of these two possibilities by means of the **exclude** attribute as follows.

```
<div type="interview">
  <u exclude="#we.sun1" xml:id="we.fun1">We had fun at the beach today.</u>
  <u exclude="#we.fun1" xml:id="we.sun1">We had sun at the beach today.</u>
</div>
```

If it is then determined that the speaker said *fun*, not *sun*, the encoder could amend the text by deleting the alternant containing *sun* and the **exclude** attribute on the remaining alternant. Alternatively, the encoder could preserve the fact that there was uncertainty in the original transcription by retaining the alternants, and assigning the **we.fun** value to the **select** attribute value on the element that encompasses the alternants, as in:

```
<div select="#we.fun2" type="interview">
  <u exclude="#we.sun2" xml:id="we.fun2">We had fun at the beach
    today.</u>
  <u exclude="#we.fun2" xml:id="we.sun2">We had sun at the beach today.</u>
</div>
```

The above alternation (including the **select** attribute) could be recoded by assigning the **exclude** attributes to tags that enclose just the words or even the characters that are mutually exclusive, as in:[12]

[12] See section *17.1. Linguistic Segment Categories* for discussion of the <w> and <c> tags that can be used in the following examples instead of the **<seg type="word">** and **<seg type="character">** tags.

```
<div type="interview">
  <u select="#fun3">We had
  <seg exclude="#sun3" xml:id="fun3" type="word">fun</seg>
    <seg exclude="#fun3" xml:id="sun3" type="word">sun</seg>
     at the beach today.</u>
</div>
```

```
<div type="interview">
  <u>We had
  <seg select="#id-f" type="word">
      <seg exclude="#id-s" xml:id="id-f" type="character">f</seg>
      <seg exclude="#id-f" xml:id="id-s" type="character">s</seg>
         un</seg>
     at the beach today.</u>
</div>
```

Now suppose that the transcriber is uncertain whether the first word in the utterance is *We* or *Lee*, but is certain that if it is *Lee*, then the other uncertain word is definitely *fun* and not *sun*. The three utterances that are in mutual exclusion can be encoded as follows.

```
<div type="interview">
<!-- ... -->
  <u exclude="#we.sun4 #lee.fun4" xml:id="we.fun4">We had fun at the beach today.</u>
  <u exclude="#we.fun4 #lee.fun4" xml:id="we.sun4">We had sun at the beach today.</u>
  <u exclude="#we.fun4 #we.sun4" xml:id="lee.fun4">Lee had fun at the beach today.</u>
<!-- ... -->
</div>
```

The preceding example can also be encoded with **exclude** attributes on the word segments *We*, *Lee*, *fun*, and *sun*:

```
<u>
  <seg exclude="#lee" xml:id="we" type="word">We</seg>
  <seg exclude="#we #sun" xml:id="lee" type="word">Lee</seg>
had
<seg exclude="#sun" xml:id="fun" type="word">fun</seg>
  <seg exclude="#fun #lee" xml:id="sun" type="word">sun</seg>
at the beach today.
</u>
```

The value of the **select** attribute is defined as a list of identifiers; hence it can also be used to narrow down the range of alternants, as in:

```
<div select="#we.fun5 #lee.fun5" type="interview">
  <u exclude="#we.sun5 #lee.fun5" xml:id="we.fun5">We had fun at the beach today.</u>
  <u exclude="#we.fun5 #lee.fun5" xml:id="we.sun5">We had sun at the beach today.</u>
  <u exclude="#we.fun5 #we.sun5" xml:id="lee.fun5">Lee had fun at the beach today.</u>
</div>
```

This is interpreted to mean that either the first or the third <u> element tag appears, and is thus equivalent to just the alternation of those two tags:

```
<div type="interview">
  <u exclude="#lee.fun6" xml:id="we.fun6">We had fun at the beach
     today.</u>
  <u exclude="#we.fun6" xml:id="lee.fun6">Lee had fun at the beach today.</u>
</div>
```

The **exclude** attribute can also be used in case there is uncertainty about the tag that appears in a certain position. For example, the occurrence of the word *May* in the s-unit *Let's go to May* can be interpreted, in the absence of other information, either as a person's name or as a date. The uncertainty can be rendered as follows, using the **exclude** attribute.

```
<s>Let's go to
<name exclude="#mayn" xml:id="mayd">May</name>
  <date copyOf="#mayd" exclude="#mayd" xml:id="mayn"/>.</s>
```

Note the use of the **copyOf** attribute discussed in section *16.6. Identical Elements and Virtual Copies*; this avoids having to repeat the content of the element whose correct tagging is in doubt.

The **copyOf** and the **exclude** attributes also provide for a simple way of indicating uncertainty about exactly where a particular element occurs in a document.[13]

For example suppose that a particular <div2> element appears either as the third and last of the <div2> elements within the first <div1> element in the body of a document, or as the first <div2> of the second <div1>. One solution would be to record the <div2> in its entirety in the first of these positions, and a virtual copy of it in the second, and mark them as excluding each other as follows:

```
<body>
  <div1 xml:id="C1">
    <div2 xml:id="C1S3" exclude="#C2S1"/>
  </div1>
  <div1 xml:id="C2">
    <div2 xml:id="C2S1" copyOf="#C1S3" exclude="#C1S3"/>
  </div1>
</body>
```

In this case, the **select** attribute, if used, would appear on the <body> element.

Mutual exclusion can also be expressed using a <link>; the first example in this section can be recoded by removing the **exclude** attributes from the <u> elements, and adding a <link> element as follows:[14]

```
<div type="interview">
  <u xml:id="we.had.fun">We had fun at the beach today.</u>
  <u xml:id="we.had.sun">We had sun at the beach today.</u>
  <link
     type="exclusiveAlternation"
     targets="#we.had.fun #we.had.sun"/>
</div>
```

Now we define the specialized linking element <alt>, making it a member of the class att.pointing, and assigning it a **mode** attribute, which can have either of the values **excl** (for exclusive) or **incl** (for inclusive). Then the following equivalence holds:

[13] An alternative way of representing this problem is discussed in chapter *21. Certainty and Responsibility*.

[14] In this example, we have placed the <link> next to the elements that represent the alternants. It could also have been placed elsewhere in the document, perhaps within a <linkGrp>.

```
    <alt mode="excl"/> = <link type="exclusive alternation"/>
```

The preceding <link> element may therefore be recoded as the following <alt> element.

```
<alt  targets="#we.had.fun #we.had.sun"  mode="excl"/>
```

Another attribute that is defined specifically for the <alt> element is **weights**, which is to be used if one wishes to assign probabilistic weights to the targets (alternants). Its value is a list of numbers, corresponding to the targets, expressing the probability that each target appears. If the alternants are mutually exclusive, then the weights must sum to 1.0.

Suppose in the preceding example that it is equiprobable whether *fun* or *sun* appears. Then the <alt> element that represents the alternation may be stated as follows:

```
<alt  targets="#we.fun #we.had.sun"  mode="excl"  weights="0.5 0.5"/>
```

The assignment of a weight of 1.0 to one target (and weights of 0 to all the others) is equivalent to selecting that target. Thus the following encoding is equivalent to the second example at the beginning of this section.

```
<u  xml:id="we.fun">We had fun at the beach today.</u>
<u  xml:id="we.sun">We had sun at the beach today.</u>
<alt  targets="#we.fun #we.sun"  mode="excl"  weights="1 0"/>
```

The sum of the weights for **<alt mode="incl">** tags ranges from 0% to (100 × k)%, where k is the number of targets. If the sum is 0%, then the alternation is equivalent to exclusive alternation; if the sum is (100 x k)%, then all of the alternants must appear, and the situation is better encoded without an <alt> tag.

If it is desired, <alt> elements may be grouped together in an <altGrp> element, and attribute values shared by the individual <alt> elements may be identified on the <altGrp> element. The **targFunc** attribute defaults to the value **first.alternant next.alternant**.

To illustrate, consider again the example of a transcribed utterance, in which it is uncertain whether the first word is *We* or *Lee*, whether the third word is *fun* or *sun*, but that if the first word is *Lee*, then the third word is *fun*. Now suppose we have the following additional information: if *we* occurs, then the probability that *fun* occurs is 50% and that *sun* occurs is 50%; if *fun* occurs, then the probability that *we* occurs is 40% and that *Lee* occurs is 60%. This situation can be encoded as follows.

```
<u>
  <seg  exclude="#lee2"  xml:id="we2"  type="word">We</seg>
  <seg  exclude="#we2"  xml:id="lee2"  type="word">Lee</seg>
had
<seg  exclude="#sun2"  xml:id="fun2"  type="word">fun</seg>
  <seg  exclude="#fun2"  xml:id="sun2"  type="word">sun</seg>
at the beach today.
</u>
<altGrp>
  <alt  targets="#we2 #lee2"/>
  <alt  targets="#fun2 #sun2"/>
  <alt  targets="#we2 #fun2"  mode="incl"  weights="0.5 0.5"/>
  <alt  targets="#lee2 #fun2"  mode="incl"  weights="1.0 0.6"/>
</altGrp>
```

As noted above, when the **mode** attribute on an <alt> has the value **incl**, then each weight states the probability that the corresponding alternative occurs, given that at least one of the other alternatives occurs.

From the information in this encoding, we can determine that the probability is about 28.5% that the utterance is 'We had fun at the beach today', 28.5% that it is *We had sun at the beach today*, and 43% that it is *Lee had fun at the beach today*.

Another very similar example is the following regarding the text of a Broadway song. In three different versions of the song, the same line reads 'Her skin is tender as a leather glove', 'Her skin is tender as a baseball glove', and 'Her skin is tender as Dimaggio's glove.'[15]

If we wish to express this textual variation using the <alt> element, we can record our relative confidence in the readings *Dimaggio's* (with probability 50%), *a leather* (25%), and *a baseball* (25%).

Let us extend the example with a further (imaginary) variation, supposing for the sake of the argument that the next line is variously given as *and she bats from right to left* (with probability 50%) or *now ain't that too damn bad* (with probability 50%). Using the <alt> element, we can express the conviction that if the first choice for the second line is correct, then the probability that the first line contains *Dimaggio's* is 90%, and each of the others 5%; whereas if the second choice for the second line is correct, then the probability that the first line contains *Dimaggio's* is 10%, and each of the others is 45%. This can be encoded, with an <altGrp> element containing a combination of exclusive and inclusive <alt> elements, as follows.

```
<div xml:id="bm" type="song">
  <l>Her skin is tender as
  <seg xml:id="dm">Dimaggio's</seg>
    <seg xml:id="lt">a leather</seg>
    <seg xml:id="bb">a baseball</seg>
     glove,</l>
  <l xml:id="rl">and she bats from right to left.</l>
  <l xml:id="db">now ain't that too damn bad.</l>
</div>
<altGrp>
  <alt targets="#dm #lt #bb" mode="excl" weights="0.5 0.25 0.25"/>
  <alt targets="#rl #db" mode="excl" weights="0.50 0.50"/>
</altGrp>
<altGrp mode="incl">
  <alt targets="#dm #rl" weights="0.90 0.90"/>
  <alt targets="#lt #rl" weights="0.5 0.5"/>
  <alt targets="#bb #rl" weights="0.5 0.5"/>
  <alt targets="#dm #db" weights="0.10 0.10"/>
  <alt targets="#lt #db" weights="0.45 0.90"/>
  <alt targets="#bb #db" weights="0.45 0.90"/>
</altGrp>
```

16.9 Stand-off Markup

16.9.1 Introduction

Most of the mechanisms defined in this chapter rely to a greater or lesser extent on the fact that tags in a marked-up document can both assert a property for a span of text which they enclose, and assert the existence of an association between themselves and some other span of text elsewhere. In stand-off markup, there is a clear separation of these two behaviours: the markup does not directly contain any part of the text, but instead includes it by reference. One specific mechanism recommended by these Guidelines for this purpose is the standard XInclude mechanism defined by the W3C; another is to use pointers as demonstrated elsewhere in this chapter.

[15] The variant readings are found in the commercial sheet music, the performance score, and the Broadway cast recording.

There are many reasons for using stand-off markup: the source text might be read-only so that additional markup cannot be added, or a single text may need to be marked up according to several hierarchically incompatible schemes, or a single scheme may need to accommodate multiple hierarchical ambiguities, so that a single markup tree is not the most faithful representation of the source material.

This section describes a generic mechanism for expressing *all* kinds of markup externally as stand-off tags, for use whenever it is appropriate.

Throughout this section the following terms will be systematically used in specific senses.

source document a document to which the stand-off markup refers (a source document can be either XML or plain text); there may be more than one source document.

internal markup markup that is already present in an XML source document

stand-off markup markup that is either outside of the source document and points in to it to the data it describes, or alternatively is in another part of the source document and points elsewhere within the document to the data it describes

external document a document that contains stand-off markup that points to a different, source document

internalize the action of creating a new XML document with external markup and data integrated with the source document data, and possibly some source document markup as well

externalize a process applied to markup from a pre-existing XML document, which splits it into two documents, an XML (external) document containing some of the markup of the original document, and another (source) XML document containing whatever text content and markup has not been extracted into the stand-off document; if all markup has been externalized from a document, the new source may be a plain text document

The three major requirements satisfied by this scheme for stand-off markup are:

1. any valid TEI markup can be either internal or external,
2. external markup can be internalized by applying it to the document content by either substituting the existing markup or adding to it, to form a valid TEI document, and
3. the external markup itself specifies whether an internalized document is to be created by substituting the existing internal markup or by adding to it

.

16.9.2 Overview of XInclude

Stand-off markup which relies on the inclusion of virtual content is adequately supported by the W3C XInclude recommendation, which is also recommended for use by these Guidelines.[16] XInclude defines a namespace (*http://www.w3.org/2001/XInclude*), which in these Guidelines will be associated with the prefix *xi:*, and exactly two elements, <xi:include> and <xi:fallback>. XInclude relies on the XPointer framework discussed elsewhere in this chapter to point to the actual fragments of text to be internalized. Although XInclude only requires support for the `element()` scheme of XPointer, these Guidelines permit the use of any of the pointing schemes discussed in section *16.2. Pointing Mechanisms*.

XInclude is a W3C recommendation which specifies a syntax for the inclusion within an XML document of data fragments placed in different resources. Included resources can be either plain text or XML. XInclude

[16] The version on which this text is based is the W3C Recommendation dated 20 December 2004..

instructions within an XML document are meant to be replaced by a resource targetted by a URI, possibly augmented by an XPointer that identifies the exact subresource to be included.

The <xi:include> element uses the **href** attribute to specify the location of the resource to be included; its value is an URI containing, if necessary, an XPointer. Additionally, it uses the **parse** attribute (whose only valid values are **text** and **xml**) to specify whether the included content is plain text or an XML fragment, and the **encoding** attribute to provide a hint, when the included fragment is text, of the character encoding of the fragment. An optional <xi:fallback> element is also permitted within an <xi:include>; it specifies alternative content to be used when the external resource cannot be fetched for some reason. Its use is not however recommended for stand-off markup.

XInclude currently only requires support for one XPointer scheme, called `element()`. As described in *16.2.3. W3C element() Scheme*, the `element()` scheme can use either a bare name (denoting an element with a specific **xml:id** attribute) or a child sequence (a numerical sequence of slash-separated child numbers specifying a path in the XML tree whose final step selects a specific subtree of XML content) to specify its target. Another scheme, `xpointer()`, has not yet become a W3C recommendation, although it has been part of the XPointer drafts from the beginning. The `xpointer()` scheme and the TEI schemes defined earlier (see *16.2.4. TEI XPointer Schemes*) add the concepts of points and ranges, which can be used to specify sub-node fragments (e.g., a few words within a longer text node) or trans-node fragments (e.g., a segment of text that spans across different branches of the overall XML tree).

16.9.3 Doing Stand-off Markup in TEI

The operations of internalizing and externalizing markup are very useful and practically important. XInclude processing as defined by the W3C *is* internalization of one or more source documents' content into a stand-off document. TEI use of XInclude for stand-off markup enables use of XInclude-conformant sofware to perform this useful operation. However, internalization is not clearly defined for all stand-off files, because the structure of the internal and external markup trees may overlap. In particular, when an external markup document selects a range that overlaps partial elements in the source document, it is not clear how the semantics of internalization (inclusion) should work, since partial elements are not XML objects.[17] XInclude defines a semantics for this case that involves only complete elements.

When a range selection partially overlaps a number of elements in a source document, XInclude specifies that the partially overlapping elements should be included as well as all completely overlapping elements and characters (partially overlapping characters are not possible). The effect of this is that elements that straddle the start or end of a selected range will be included as wrappers for those of their children that are completely or partially selected by the range. For example, given the following source document:

```
<body>
<p xml:id="par1">home, <emph>home</emph> on Brokeback Mountain.</p>
<p xml:id="par2">That was the <emph>song</emph> that I sang</p>
</body>
```

and the following external document:

```
<body xmlns:xi="http://www.w3.org/2001/XInclude">
<div><xi:include href="example1.xml"
xpointer="range(xpath1(id(par1)//emph),xpath1(id(par2)//emph))"/>
</div>
```

[17] This corresponds to the observation that overlapping XML tags reflecting a textual version of such an inclusion would not even be well-formed XML. This kind of overlap in textual phenomena of interest is in fact the major reason that stand-off markup is needed.

```
    </body>
```

the resulting document after XInclude processing of this external document would be:

```
    <body xmlns:xi="http://www.w3.org/2001/XInclude">
    <div>
    <p xml:id="par1"><emph>home</emph> on Brokeback Mountain.</p>
    <p xml:id="par2">That was the <emph>song</emph></p>
    </div>
    </body>
```

The result of the inclusion is two paragraph elements, while the original range designated in the source document overlapped two paragraph fragments.

The semantics of XInclude require the creation of well-formed XML results even though the pointing mechanisms it uses do not necessarily respect the hierarchical structure of XML documents, as in this case. While this is a good way to ensure that internalization is always possible, it has implications for the use of XInclude as a notation for the *description* of overlapping markup structures.

When overlapping hierarchies need to be represented for a single document, each hierarchy must be represented by a separate set of XInclude tags pointing to a common source document. This sort of structure corresponds to common practice in work with linguistic text corpora. In such corpora, each potentially overlapping hierarchy of elements for the text is represented as a separate stream of stand-off markup. Generally the source text contains markup for the smallest significant units of analysis in the corpus, such as words or morphemes, this information and its markup representing a layer of common information that is shared by all the various hierarchies. As a way of organizing the representation of complex data, this technique generally allows a large number of **xml:id** attributes to be attached to the shared elements, providing robust anchors for links and facilitating adjustments to the source document without breaking external documents that reference it.

Any tag can be externalized by

removing its content and replacing it with an <xi:include> element that contains an XPointer pointing to the desired content.

For instance the following portion of a TEI document:

```
<text>
  <body>
    <head>1755</head>
    <l>To make a prairie it takes a clover and one bee,</l>
    <l>One clover, and a bee,</l>
    <l>And revery.</l>
    <l>The revery alone will do,</l>
    <l>If bees are few.</l>
  </body>
</text>
```

Source: [60]

can be externalized by placing the actual text in a separate document, and providing exactly the same markup with the <xi:include> elements:

```
<content>To make a prairie it takes a clover and one bee,\n
One clover, and a bee,\n
And revery.\n
The revery alone will do,\n
If bees are few.\n
</content>
```

```
<text xmlns:xi="http://www.w3.org/2001/XInclude">
<body>
<head>1755</head>
<l>
<xi:include href="Source.xml" parse="xml"
xpointer="string-range(element(/1),  0, 48)"/>
</l>
<l>
<xi:include href="Source.xml" parse="xml"
xpointer="string-range(element(/1), 49, 71)"/>
</l>
<l>
<xi:include href="Source.xml" parse="xml"
xpointer="string-range(element(/1), 72, 83)"/>
</l>
<l>
<xi:include href="Source.xml" parse="xml"
xpointer="string-range(element(/1), 84,109)"/>
</l>
<l>
<xi:include href="Source.xml" parse="xml"
xpointer="string-range(element(/1),110,126)"/>
</l>
</body>
</text>
```

Please note that this specification requires that the XInclude namespace declaration is present in all cases. The <xi:fallback> element contains text or XML fragments to be placed in the document if the inclusion fails for any reason (for instance due to inaccessibility of an external resource). The <xi:fallback> element is optional; if it is not present an XInclude processor must signal a fatal error when a resource is not found. This is the preferred behaviour for use with stand-off markup. These Guidelines recommend against the use of <xi:fallback> for stand-off markup.

16.9.4 Well-formedness and Validity of Stand-off Markup

The whole source fragment identified by an XInclude element, as well as any markup therein contained is inserted in the position specified, and an XInclude processor is required to ensure that the resulting internalized document is well-formed. This has obvious implications when the external document contains XML markup. A plain text source document will always create a well-formed internalized document.

While a TEI customization may permit <xi:include> elements in various places in a TEI document instance, in general these Guidelines suggest that validity be verified after the resolution of all the <xi:include> elements.

16.9.5 Including Text or XML Fragments

When the source text is plain text the overall form of the XPointer pointing to it is of minimal importance. The form of the XPointer matters considerably, on the other hand, when the source document is XML.

In this case, it is rather important to distinguish whether we intend to substitute the source XML with the new one, or just to add new markup to it. The XPointers used in the references can express both cases.

A simple way is to make sure to select only textual data in the XPointer. For instance, given the following document:

```
<html>
<body>
<div>To make a prairie it takes a <a href="clover.gif">clover</a>
and one <a href="bee.gif">bee</a>,</div>
<div>One <a href="clover.gif">clover</a>, and
a <a href="bee.gif">bee</a>,</div>
<div>And revery.</div>
<div>The revery alone will do,</div>
<div>If bees are few.</div>
</body>
</html>
```

the expression `range(/1/2/1.0,/1/2/11.1)` will select the whole poem, text content *and* elements *and* hypertext links (NB: in XPointer whitespace-only text nodes count).

On the contrary, the expressions `xpointer(//text()/range-to(.))` and `xpointer(string-range(//text(),"To")/range-to(//text(),"few.")` will only select the text of the poem, with no markup inside.

Thus, the following could be a valid stand-off document for the *Source.xhtml* document:

```
<text xmlns:xi="http://www.w3.org/2001/XInclude">
<body>
<head>1755</head>
<l>
<xi:include href="Source.xhtml"
xpointer='xpointer(string-range(//div[1]/text(),"To")/range-to(//div[1]/text(),"bee,")'/>
</l>
<l>
<xi:include href="Source.xhtml"
xpointer='xpointer(string-range(//div[2]/text(),"One")/range-to(//div[2]/text(),"bee,")'/>
</l>
<l>
<xi:include href="Source.xhtml"
xpointer='xpointer(string-range(//div[3]/text(),"And")/range-to(//div[3]/text(),".")'/>
</l>
<l>
<xi:include href="Source.xhtml"
xpointer='xpointer(string-range(//div[4]/text(),"The")/range-to(//div[4]/text(),",")'/>
</l>
<l>
<xi:include href="Source.xhtml"
xpointer='xpointer(string-range(//div[5]/text(),"If")/range-to(//div[5]/text(),".")'/>
</l>
```

```
</body>
</text>
```

16.10 Connecting Analytic and Textual Markup

In chapters *17. Simple Analytic Mechanisms* and *18. Feature Structures* and elsewhere, provision is made for analytic and interpretive markup to be represented outside of textual markup, either in the same document or in a different document. The elements in these separate domains can be connected, either with the pointing attributes **ana** (for *analysis*) and **inst** (for *instance*), or by means of <link> and <linkGrp> elements. Numerous examples are given in these chapters.

16.11 Module for Linking, Segmentation, and Alignment

The module described in this chapter makes available the following components:

Module linking: Linking, segmentation and alignment

- Elements defined: ab alt altGrp anchor join joinGrp link linkGrp seg timeline when
- Classes defined: att.global.linking att.pointing att.pointing.group

The selection and combination of modules to form a TEI schema is described in *1.2. Defining a TEI Schema*.

Chapter 17

Simple Analytic Mechanisms

This chapter describes a module for associating simple analyses and interpretations with text elements. We use the term analysis here to refer to any kind of semantic or syntactic interpretation which an encoder wishes to attach to all or part of a text. Examples discussed in this chapter include familiar linguistic categorizations (such as 'clause', 'morpheme', 'part-of-speech' etc.) and characterizations of narrative structure (such as 'theme', 'reconciliation' etc.). The mechanisms presented in this chapter are simpler but less powerful than those described in chapter *18. Feature Structures*.

Section *17.1. Linguistic Segment Categories* introduces a module for characterizing text segments according to the familiar linguistic categories of sentence or s-unit, clause, phrase, word, morpheme, and character. These elements represent special cases of the generic <seg> element described in section *16.3. Blocks, Segments, and Anchors*.

Section *17.2. Global Attributes for Simple Analyses* introduces an additional global attribute which allows passages of text to be associated with specialised elements representing their interpretation. These 'interpretative' elements (and <interp>) are described in detail in section *17.3. Spans and Interpretations*. They allow the encoder to specify an analysis as a series of names and associated values,[1] each such pair being linked to one or more stretches of text, either directly, in the case of spans, or indirectly, in the case of interpretations.

Finally section *17.4. Linguistic Annotation* revisits the topic of linguistic analysis, and illustrates how these interpretative mechanisms may be used to associate simple linguistic analysis with text segments.

17.1 Linguistic Segment Categories

In this section we introduce specialized linguistic segment category elements which may be used to represent the segmentation of a text into the traditional linguistic categories of sentence, clause, phrase, word, morpheme, and characters.

<s> (s-unit) contains a sentence-like division of a text.

<cl> (clause) represents a grammatical clause.

<phr> (phrase) represents a grammatical phrase.

<w> (word) represents a grammatical (not necessarily orthographic) word.
@lemma provides a lemma for the word, such as an uninflected dictionary entry form.

<m> (morpheme) represents a grammatical morpheme.
@baseForm identifies the morpheme's base form.

<c> (character) represents a character.

As members of the att.segLike class, these elements all share the following attributes:

[1] Or, as they are widely known, attribute-value pairs; this term should not be confused, however, with SGML or XML attributes and their values, which are similar in concept but distinct in their formal definitions.

att.segLike provides attributes for elements used for arbitrary segmentation.

@**type** characterizes the type of segment.

@**function** characterizes the function of the segment.

These elements are also all members of the model.segLike class, which is a subclass of model.phrase. They may thus appear anywhere that text is permitted within a document, when the module defined by this chapter is included in a schema.

The <s> element may be used simply to segment a text end-to-end into a series of non-overlapping segments, referred to here and elsewhere as s-units, or sentences.

```
<p>
  <s>Nineteen fifty-four, when I was eighteen years old,
     is held to be a crucial turning point in the history of
     the Afro-American — for the U.S.A. as a whole — the
     year segregation was outlawed by the U.S. Supreme Court.</s>
  <s>It was also a crucial year for me because on June 18,
     1954, I began serving a sentence in state prison for
     possession of marijuana.</s>
</p>
```

Source: [37]

The <s> element is more restricted both in its content and its usage than the generic <seg> element. The <seg> unit may contain anything which can appear within a paragraph: thus it may be used to enclose members of the model.inter class (such as <bibl> or <list>) as well as other phrase elements; the <s> unit may only contain phrase-level elements or text. Also, unlike <seg> elements, <s> elements should not be nested within each other.[2] The <seg> element is intended for use as a generic segmentation element, the specific function of which may be indicated by its **type** attribute; the other members of the class are more specialised. Thus, the <s>, <cl>, and <phr> elements may be thought of as equivalent to **<seg type="s-unit">**, **<seg type="clause">** and **<seg type="phrase">**, respectively, but with the above-mentioned restrictions.

The <s> element may be further subdivided into clauses, marked with the <cl> element, as in the following example:

```
<p>
  <s>
    <cl>It was about the beginning of September, 1664,
    <cl>that I, among the rest of my neighbours,
           heard in ordinary discourse
      <cl>that the plague was returned again to Holland; </cl>
      </cl>
    </cl>
    <cl>for it had been very violent there, and particularly at
        Amsterdam and Rotterdam, in the year 1663, </cl>
    <cl>whither, <cl>they say,</cl> it was brought,
    <cl>some said</cl> from Italy, others from the Levant, among some goods
    <cl>which were brought home by their Turkey fleet;</cl>
    </cl>
    <cl>others said it was brought from Candia;
        others from Cyprus. </cl>
  </s>
  <s>
```

[2] Neither this constraint, nor the requirement that the whole of the text be segmented by <s> elements is enforced by the current TEI schemas; such constraints may however be introduced in a later version of these Guidelines.

```
   <cl>It mattered not <cl>from whence it came;</cl>
   </cl>
   <cl>but all agreed <cl>it was come into Holland again.</cl>
   </cl>
  </s>
</p>
```

Source: [55]

Clauses may be further divided into <phr> elements in the same way. A text may be segmented directly into clauses, or into phrases, with no need to include segmentation at a higher level as well.

For verse texts, the overlapping of metrical and syntactic structure requires that special care be given to representing both using an element hierarchy. One simple approach is to split the syntactic phrases into fragments when they cross verse boundaries, reuniting them with the **part** attribute:

```
<div type="stanza">
  <l>
    <cl part="I">Tweedledum and Tweedledee</cl>
  </l>
  <l>
    <cl part="F">Agreed to have a battle;</cl>
  </l>
  <l>
    <cl part="I">For Tweedledum said <cl part="I">Tweedledee</cl>
    </cl>
  </l>
  <l>
    <cl part="F">
      <cl part="F">Had spoiled his nice new rattle.</cl>
    </cl>
  </l>
</div>
<div type="stanza">
  <l>
    <cl part="I">Just then flew down a monstrous crow,</cl>
  </l>
  <l>
    <cl part="F">As black as a tar barrel;</cl>
  </l>
  <l>
    <cl part="I">Which frightened both the heroes so,</cl>
  </l>
  <l>
    <cl part="F">
      <cl>They quite forgot their quarrel.</cl>
    </cl>
  </l>
</div>
```

Source: [30]

Another approach is to use the **next** and **prev** attributes defined in the additional module for linking (chapter *16. Linking, Segmentation, and Alignment*):

```
<l>
  <cl next="#c5" xml:id="c3" part="I">For Tweedledum said
  <cl next="#c6" xml:id="c4" part="I">Tweedledee</cl>
  </cl>
</l>
<l>
  <cl prev="#c3" xml:id="c5" part="F">
    <cl prev="#c4" xml:id="c6" part="F">Had spoiled his nice new rattle.</cl>
  </cl>
</l>
```

Other methods are also possible; for discussion, see chapter *20. Non-hierarchical Structures*.

The **type** attribute on linguistic segment categories can be used to provide additional interpretative information about the category. The **function** attribute on the <cl> and <phr> elements can be used to provide additional information about the function of the category. Legal values for these two attributes are not defined by these Guidelines, but should be documented in the <segmentation> element of the <encodingDesc> element within the document's header. A general approach to the encoding of linguistic categories for parts of a text is discussed in section *17.4. Linguistic Annotation* below.

Using traditional terminology, these attributes provide a convenient way of specifying, for example, that the clause *from whence it came* is a relative clause modifying another, or that the phrase *by the U.S. Supreme Court* is a prepositional post-modifier:

```
<cl>It mattered not
<cl type="relative" function="clause_modifier">from whence it came;</cl>
</cl>
```

```
<phr type="NP">the year segregation</phr>
<phr>was outlawed</phr>
<phr type="PP" function="postmodifier-agent">by the U.S. Supreme Court.</phr>
```

Segmentation into clauses and phrases can, of course, be combined. Such detailed encodings as the following may require careful formatting if they are to be easily readable however.

```
<p>
  <s>
    <cl type="finite-declarative" function="independent">
      <phr type="NP" function="subject">Nineteen fifty-four,
      <cl type="finite-relative-declarative" function="appositive">when                <phr type="NP" func-
tion="subject">I</phr>
          <phr type="VP" function="predicate">was eighteen years old</phr>
        </cl>
      </phr>,
    <phr type="VP" function="predicate">
       <phr type="V" function="verb-main">is held</phr>
       <phr type="NP" function="complement">
         <cl type="nonfinite" function="predicate-nom.">
           <phr type="V" function="copula">to be</phr>
           <phr type="NP" function="predicate-nom.">a crucial turning point
           <phr type="PP" function="postmodifier">in
             <phr type="NP" function="prep.obj.">the history
```

```
            <phr type="PP" function="postmodifier">of the Afro-American</phr>
            </phr>
          </phr>
                    –
          <phr type="PP" function="postmodifier-appositive">for
            <phr type="NP" function="prep.obj.">the U.S.A.
              <phr type="PP" function="postmodifier">as a whole</phr>
              </phr>
            </phr>
          </phr>
                  –
        <phr type="NP" function="appositive-predicate-nom.">the year
          <cl type="finite-relative" function="adjectival">
            <phr type="NP" function="subject">segregation</phr>
            <phr type="VP" function="predicate">
              <phr type="V" function="verb-main">was outlawed</phr>
              <phr type="PP" function="postmodifier">by the U.S. Supreme Court</phr>
            </phr>
          </cl>
          </phr>
        </cl>
      </phr>
    </phr>.</cl>
  </s>
  <s>
    <cl type="finite-declarative" function="independent">
      <phr type="NP" function="subject">It</phr>
      <phr type="VP" function="predicate">
        <phr type="V" function="verb-main">was</phr>
            also
      <phr type="NP" function="predicate-nom.">a crucial year for me</phr>
      </phr>
      <cl type="declarative-finite" function="dependent-causative">because
      <phr type="PP" function="sentence_adverb">on June 18, 1954</phr>,
      <phr type="NP" function="subject">I</phr>
        <phr type="VP" function="predicate">
          <phr type="V" function="verb-main">began serving</phr>
          <phr type="NP" function="complement">a sentence in state prison
          <phr type="PP" function="complement">for possession of marijuana</phr>
          </phr>
        </phr>
      </cl>
    </cl>
  </s>.
</p>
```

This style of markup may introduce spurious new lines and blanks into the text. If the original layout is important, it should be explicitly encoded, using such facilities as the <lb> element, the global **rend** or **rendition** attributes, etc.

The <w>, <m>, and <c> elements are also identical in meaning to the <seg> element with a **type** attribute of 'w', 'm', or 'c', and may occur wherever <seg> is permitted to occur. However, they have more restricted content models than does <seg>: for example, the <w> element should only contain <w>, <m>, and <c> elements, or plain text; the <m> element should contain only <c> elements or plain text; the <c> element should contain only plain text, most often only a single character or a sequence of graphemes to be treated as a single character.

Consequently, while these more specific elements can be translated directly into typed <seg> elements, the reverse is not necessarily the case.

The restriction on the content of the <w> attribute in particular requires that a certain care must be exercised when using it, especially in relation to the use of other tags that one may think of as word level, but which are in fact defined as phrase level. Consider the problem of segmenting an occurrence of the <mentioned> element as a word.

```
<mentioned>grandiloquent</mentioned>
```

The first of the following two encodings is legitimate; the second is not, since the <mentioned> element is not part of the content model of the <w> element:

```
<mentioned>
  <w>grandiloquent</w>
</mentioned>
```

```
    <w><mentioned>grandiloquent</mentioned></w>
```

On the other hand, both of the following encodings *are* legitimate:

```
<mentioned>
  <phr>grandiloquent speech</phr>
</mentioned>
```

```
<phr>
  <mentioned>grandiloquent speech</mentioned>
</phr>
```

The first encoding describes the citing of a phrase. The second describes a phrase which consists of something mentioned.

The <w> and <m> elements carry additional attributes which may be of use in many indexing or analytic applications. The **lemma** attribute may be used to specify the lemma, that is the head- or base- form of an inflected verb or noun, for example:

```
<s  xml:lang="la">
  <w  lemma="timeo">timeo</w>
  <w  lemma="danaii">Danaos</w>
  <w  lemma="et">et</w>
  <w  lemma="donum">dona</w>
  <w  lemma="fero">ferentes</w>
</s>
```

Similarly, the **baseForm** attribute may be specified for the <m> element, to indicate the 'base form' of a transformed morpheme:

```
<w type="adjective">
  <m type="prefix" baseForm="con">com</m>
  <m type="root">fort</m>
  <m type="suffix">able</m>
</w>
```

The <w>, <m>, and <c> elements can be used together to give a fairly detailed low-level grammatical analysis of text. For example, consider the following segmentation of the English S-unit *I didn't do it*.

```
<w>I</w>
<w>
  <w>did</w>
  <m>n't</m>
</w>
<w>do</w>
<w>it</w>
<c>.</c>
```

This segmentation, crude as it is, succeeds in representing the idea that *did* occurs as a word inside the word *didn't*. A further advantage of segmenting the text down to this level is that it becomes relatively simple to associate each such segment with a more detailed formal analysis. This matter is taken up in detail in section *17.4. Linguistic Annotation*.

17.2 Global Attributes for Simple Analyses

When the module described by this chapter is selected, an additional attribute is defined for all elements:

att.global.analytic provides additional global attributes for associating specific analyses or interpretations with appropriate portions of a text.

@**ana** (analysis) indicates one or more elements containing interpretations of the element on which the **ana** attribute appears.

The **ana** attribute may be specified for any element. Its effect is to associate the element with one or more others representing an analysis or interpretation of it. Its target should be one of the elements described in the section *17.3. Spans and Interpretations* below, or some other interpretative element such as <note>, on which see section *3.8. Notes, Annotation, and Indexing* or <fs>, on which see chapter *18. Feature Structures*.

17.3 Spans and Interpretations

The simplest mechanisms for attaching analytic notes in some structured vocabulary to particular passages of text are provided by the and <interp> elements, and their associated grouping elements <spanGrp> and <interpGrp>.

**** associates an interpretative annotation directly with a span of text.

<spanGrp> (span group) collects together span tags.

<interp> (interpretation) summarizes a specific interpretative annotation which can be linked to a span of text.

<interpGrp> (interpretation group) collects together a set of related interpretations which share responsibility or type.

These elements are all members of the class att.interpLike, and thus share the following attributes:

att.interpLike provides attributes for elements which represent a formal analysis or interpretation.

@**resp** (responsible party) indicates who is responsible for the interpretation.

@**type** indicates what kind of phenomenon is being noted in the passage.

@**inst** (instances) points to instances of the analysis or interpretation represented by the current element.

The **type** attribute of the and <interp> elements may be used to indicate that the annotations are of specific types, for example thematic or structural. The annotation itself is supplied as the content of the or <interp> element. In the case of the element, the span of text being annotated is indicated by values of the **from** and **to** attributes, the value of each being a pointer. If the optional **to** attribute is omitted, the span consists just of the element pointed at by the obligatory **from** attribute. In the case of <interp> (see below), the span is indicated by a pointer from a <link> element or some similar mechanism. The **resp** attribute indicates the annotator responsible for this annotation. Here is an example of the element.

```
<p xml:id="MaQp1s2p114">
  <s xml:id="MaQp1s2p114s1">There was certainly a definite point at which the
    thing began.</s>
  <s xml:id="MaQp1s2p114s2">It was not; then it was suddenly inescapable,
    and nothing could have frightened it away.</s>
  <s xml:id="MaQp1s2p114s3">There was a slow integration, during which she,
    and the little animals, and the moving grasses, and the sun-warmed
    trees, and the slopes of shivering silvery mealies, and the great
    dome of blue light overhead, and the stones of earth under her feet,
    became one, shuddering together in a dissolution of dancing
    atoms.</s>
  <s xml:id="MaQp1s2p114s4">She felt the rivers under the ground forcing
    themselves painfully along her veins, swelling them out in an
    unbearable pressure; her flesh was the earth, and suffered growth
    like a ferment; and her eyes stared, fixed like the eye of the
    sun.</s>
  <s xml:id="MaQp1s2p114s5">Not for one second longer (if the terms for time
    apply) could she have borne it; but then, with a sudden movement
      forwards and out, the whole process stopped; and <emph rend="italic">that</emph> was
<soCalled rend="dquo">the
        moment</soCalled> which it was impossible to remember
    afterwards.</s>
  <span from="#MaQp1s2p114s3" to="#MaQp1s2p114s5">the moment</span>
  <s xml:id="MaQp1s2p114s6">For during that space of time (which was
    timeless) she understood quite finally her smallness, the
    unimportance of humanity.</s>
</p>
```

Source: [123]

The element may, as in this example, be placed in the text near the textual span it is associated with. Alternatively, it may be placed elsewhere in the same or a different document. Where several or <interp> elements share the same attributes, for example having the same responsibility or type, it may be convenient to group them within a <spanGrp> or <interpGrp> element as follows:

```
<spanGrp resp="#DTL">
  <span from="#MaQp1s2p114s3" to="#MaQp1s2p114s5">the moment</span>
<!-- other spans identified by DTL here -->
</spanGrp>
```

Spans may also be used to represent structural divisions within a narrative, particularly when these do not coincide with the structure implied by the element structure. Consider the following narrative:

> Sigmund, the son of Volsung, was a king in Frankish country. Sinfiotli was the eldest of his sons, the second was Helgi, the third Hamund. Borghild, Sigmund's wife, had a brother named — But Sinfiotli, her stepson, and — both wooed the same woman and Sinfiotli killed him over it.[3] And when he came home, Borghild asked him to go away, but Sigmund offered her weregild, and she was obliged to accept it. At the funeral feast Borghild was serving beer. She took poison, a big drinking horn full, and brought it to Sinfiotli. When Sinfiotli looked into the horn, he saw that poison was in it, and said to Sigmund 'This drink is cloudy, old man.' Sigmund took the horn and drank it off. It is said that Sigmund was hardy and that poison did him no harm, inside or out. And all his sons could tolerate poison on their skin. Borghild brought another horn to Sinfiotli, and asked him to drink, and everything happened as before. And a third time she brought him a horn, and reproachful words as well, if he didn't drink from it. He spoke again to Sigmund as before. He said 'Filter it through your mustache, son!' Sinfiotli drank it off and at once fell dead.
>
> Sigmund carried him a long way in his arms and came to a long, narrow fjord, and there was a small boat there and a man in it. He offered to ferry Sigmund over the fjord. But when Sigmund carried the body out to the boat, it was fully laden. The man said Sigmund should go around the fjord inland. The man pushed the boat out and then suddenly vanished.
>
> King Sigmund lived a long time in Denmark in the kingdom of Borghild, after he married her. Then he went south to Frankish lands, to the kingdom he had there. Then he married Hiordis, the daughter of King Eylimi. Their son was Sigurd. King Sigmund fell in a battle with the sons of Hunding. And then Hiordis married Alf, the son of King Hialprec. Sigurd grew up there as a boy.
>
> Sigmund and all his sons were tall and outstanding in their strength, their growth, their intelligence, and their accomplishments. But Sigurd was the most outstanding of all, and everyone who knows about the old days says he was the most outstanding of men and the noblest of all the warrior kings.

A structural analysis of this text, dividing it into narrative units in a pattern shared with other texts from the same literature, might look like this:

```
<p xml:id="P1">
  <s xml:id="S1">Sigmund ... was a king in Frankish country.</s>
  <s xml:id="S2">Sinfiotli was the eldest of his sons.</s>
  <s xml:id="S3">Borghild, Sigmund's wife, had a brother ...</s>
  <s xml:id="S4A">But Sinfiotli ... wooed the same woman</s>
  <s xml:id="S4B">and Sinfiotli killed him over it.</s>
  <s xml:id="S5">And when he came home, ... she was obliged to accept it.</s>
  <s xml:id="S6">At the funeral feast Borghild was serving beer.</s>
  <s xml:id="S7">She took poison ... and brought it to Sinfiotli.</s>
  <s xml:id="S17">Sinfiotli drank it off and at once fell dead.</s>
  <anchor xml:id="EOS17"/>
</p>
<p xml:id="P2">Sigmund carried him a long way in his arms ... </p>
<p xml:id="P3">King Sigmund lived a long time in Denmark ... </p>
<p xml:id="P4">Sigmund and all his sons were tall ... </p>
<spanGrp resp="#TMA" type="narrative-structure">
  <span from="#S1" to="#S3">introduction</span>
  <span from="#S4A">conflict</span>
```

[3] The rule marks spaces left for the missing name in the manuscript.

```
  <span  from="#S4B">climax</span>
  <span  from="#S5"  to="#S17">revenge</span>
  <span  from="#EOS17">reconciliation</span>
  <span  from="#P2"  to="#P4">aftermath</span>
</spanGrp>
```

Source: [167]

Note the use of an empty <anchor> element to provide a target for the 'reconciliation' unit which is normally part of the narrative pattern but which is not realized in the text shown.

The same analysis may be expressed with the <interp> element instead of the element; this element provide attributes for recording an interpretive category and its value, as well as the identity of the interpreter, but does not itself indicate which passage of text is being interpreted; the same interpretive structures can thus be associated with many passages of the text. The association between text passages and <interp> elements must be made either by pointing from the text to the <interp> element with the **ana** attribute defined in section *17.2. Global Attributes for Simple Analyses*, or by pointing at both text and interpretation from a <link> element, as described in chapter *16. Linking, Segmentation, and Alignment*.

To encode the first example above using <interp>, it is necessary to create a text element which contains — or corresponds to — the third, fourth, and fifth orthographic sentences (S-units) in the paragraph. This can be done either with the <seg> element, described in *16.3. Blocks, Segments, and Anchors*, or the <join> element, described in *16.7. Aggregation*. The resulting element can then be associated with the <interp> element using the **ana** attribute described in section *17.2. Global Attributes for Simple Analyses*. We illustrate using the <seg> element.

```
<p  xml:id="MarQp1s2p114">
  <s  xml:id="MarQp1s2p114s1">There was certainly a definite point ... </s>
  <s  xml:id="MarQp1s2p114s2">It was not; then it was suddenly inescapable ... </s>
  <seg  xml:id="MarQp1s2p114s3-5"  ana="#moment">
    <s  xml:id="MarQp1s2p114s3">There was a slow integration ... </s>
    <s  xml:id="MarQp1s2p114s4">She felt the rivers under the ground ... </s>
    <s  xml:id="MarQp1s2p114s5">Not for one second longer ... </s>
  </seg>
  <s  xml:id="MarQp1s2p114s6">For during that space of time ... </s>
</p>
<interp  xml:id="moment">the moment</interp>
```

The second example above can be recoded using <interp> and <interpGrp> tags in a similar manner. The interpretation itself can be expressed in an <interpGrp> element, which would replace the <spanGrp> in the example shown above:

```
<interpGrp  resp="#TMA"  type="structuralUnit">
  <interp  xml:id="INTRO">introduction</interp>
  <interp  xml:id="CONFLICT">conflict</interp>
  <interp  xml:id="CLIMAX">climax</interp>
  <interp  xml:id="REVENGE">revenge</interp>
  <interp  xml:id="RECONCIL">reconciliation</interp>
  <interp  xml:id="AFTERM">aftermath</interp>
</interpGrp>
```

Any of these <interp> elements may be linked to the text either by means of the **ana** attribute, or by means of <link> elements. Using the **ana** attribute (on <seg> elements introduced specifically for this purpose), the text would be encoded as follows:

```
<p xml:id="PP1">
  <seg xml:id="SS1-SS3" ana="#INTRO">
    <s xml:id="SS1">Sigmund ... was a king in Frankish country.</s>
    <s xml:id="SS2">Sinfiotli was the eldest of his sons.</s>
    <s xml:id="SS3">Borghild, Sigmund's wife, had a brother ... </s>
  </seg>
  <s xml:id="SS4A" ana="#CONFLICT">But Sinfiotli ... wooed the same woman</s>
  <s xml:id="SS4B" ana="#CLIMAX">and Sinfiotli killed him over it.</s>
  <seg xml:id="SS5-SS17" ana="#REVENGE">
    <s xml:id="SS5">And when he came home, ... she was obliged to accept it.</s>
    <s xml:id="SS6">At the funeral feast Borghild was serving beer.</s>
    <s xml:id="SS17">Sinfiotli drank it off and at once fell dead.</s>
  </seg>
</p>
<anchor xml:id="NIL1" ana="#RECONCIL"/>
<p xml:id="PP2">Sigmund carried him a long way in his arms ... </p>
<p xml:id="PP3">King Sigmund lived a long time in Denmark ... </p>
<p xml:id="PP4">Sigmund and all his sons were tall ... </p>
<join xml:id="PP2-PP4" targets="#PP2 #PP3 #PP4" ana="#AFTERM"/>
```

The linkage may also be accomplished using a <linkGrp> element, whose content is a set of <link> elements which point to each interpretive element and its corresponding text unit. This method does not require the use of the **ana** attribute on the text units.

```
<linkGrp targFunc="interpretation text">
  <link targets="#INTRO #SS1-SS3"/>
  <link targets="#CONFLICT #SS4A"/>
  <link targets="#CLIMAX #SS4B"/>
  <link targets="#REVENGE #SS5-SS17"/>
  <link targets="#RECONCIL #NIL1"/>
  <link targets="#AFTERM #PP2-PP4"/>
</linkGrp>
```

One obvious advantage of using <interp> rather than elements for the Sigmund text is that the <interp> elements can be reused for marking up other texts in the same document, whereas the elements cannot. Another is that the <interp> element can be used to provide interpretations for discontinuous text elements (represented by <join> elements). On the other hand, the use of <interp> elements may require the creation of special text elements not otherwise needed (e.g. the <seg> and the <join> in the revised encoding of the text), whereas the use of elements does not.

17.4 Linguistic Annotation

By linguistic annotation we mean here any annotation determined by an analysis of linguistic features of the text, excluding as borderline cases both the formal structural properties of the text (e.g. its division into chapters or paragraphs) and descriptive information about its context (the circumstances of its production, its genre or medium). The structural properties of any TEI-conformant text should be represented using the structural elements discussed elsewhere in this chapter and in chapters *3. Elements Available in All TEI Documents*, *4. Default Text Structure*, and the various chapters of Part III. The contextual properties of a TEI text are fully documented in the TEI Header, which is discussed in chapter *2. The TEI Header*, and in section *15.2. Contextual Information*.

Other forms of linguistic annotation may be applied at a number of levels in a text. A code (such as a word-class or part-of-speech code) may be associated with each word or token, or with groups of such tokens,

which may be continuous, discontinuous, or nested. A code may also be associated with relationships (such as cohesion) perceived as existing between distinct parts of a text. The codes themselves may stand for discrete and non-decomposable categories, or they may represent highly articulated bundles of textual features. Their function may be to place the annotated part of the text somewhere within a narrowly linguistic or discoursal domain of analysis, or within a more general semantic field, or any combination drawn from these and other domains.

The manner by which such annotations are generated and attached to the text may be entirely automatic, entirely manual or a mixture. The ease and accuracy with which analysis may be automated may vary with the level at which the annotation is attached. The method employed should be documented in the <interpretation> element within the encoding description of the TEI Header, as described in section *2.3.3. The Editorial Practices Declaration*. Where different parts of a language corpus have used different annotation methods, the **decls** attribute may be used to indicate the fact, as further discussed in section *15.3. Associating Contextual Information with a Text*.

As one example of such types of analysis, consider the following sentence, taken from the Lancaster/IBM Treebank Project (Leech and Garside (1991)).

> The victim's friends told police that Kruger drove into the quarry and never surfaced.

Our discussion focuses on the way that this sentence might be analysed using the CLAWS system developed at the University of Lancaster but exactly the same principles may be applied to a wide variety of other systems.[4] Output from the system consists of a segmented and tokenized version of the text, in which word class codes have been associated with each token. CLAWS offers outputs in a variety of non-XML and XML formats: for example, the simplest format for the sample sentence would be:

```
The_AT0 victim_NN1 's_POS friends_NN2 told_VVD police_NN2 that_CJT Kruger_NP0
drove_VVD into_PRP the_AT0 quarry_NN1 and_CJC never_AV0 surfaced_VVD
```

This may be easily transformed into an equivalent TEI XML representation:

```
<s>
  <w ana="#AT0">The </w>
  <w ana="#NN1">victim</w>
  <w ana="#POS">'s</w>
  <w ana="#NN2">friends </w>
  <w ana="#VVD">told </w>
  <w ana="#NN2">police </w>
  <w ana="#CJT">that </w>
  <w ana="#NP0">Kruger </w>
  <w ana="#VVD">drove </w>
  <w ana="#PRP">into </w>
  <w ana="#AT0">the </w>
  <w ana="#NN1">quarry </w>
  <w ana="#CJC">and </w>
  <w ana="#AV0">never </w>
  <w ana="#VVD">surfaced</w>
</s>
```

Although the names used for the attribute values here may have some significance for the human reader (**AT0** for article, **NN1** for singular noun, **NN2** for plural noun, etc.) they are arbitrary codes, used in this case

[4] For the word-class tagging method used by CLAWS see Marshall (1983); For an overview of the system see Garside et al. (1991). The example sentence was processed using an online version of the CLAWS tagger at `http://www.comp.lancs.ac.uk/ucrel/claws/trial.html`

as pointers to other elements which define their significance more precisely. If the codes are considered to be atomic, then the <interp> element described in section *17.3. Spans and Interpretations* might be used to supply brief definitions in the header:

```
<interpGrp type="POS">
  <interp xml:id="AT0">Definite article</interp>
  <interp xml:id="AV0">Adverb</interp>
  <interp xml:id="CJC">Conjunction</interp>
  <interp xml:id="CJT">Relative that</interp>
  <interp xml:id="NN1">Noun singular</interp>
  <interp xml:id="NN2">Noun plural</interp>
  <interp xml:id="NP0">Proper noun</interp>
  <interp xml:id="POS">Genitive marker</interp>
  <interp xml:id="PRP">Preposition</interp>
  <interp xml:id="VVD">Verb past tense</interp>
</interpGrp>
```

If the codes are considered to be compositional (for example that **NN1** and **NN2** have something in common, namely their noun-ness, which they do not share with, say, **VVD**), then this compositionality may be most clearly expressed using a mechanism based on the <fs> element defined in chapter *18. Feature Structures*.

This approach requires the text to be fully segmented, using the linguistic segment elements described in section *17.1. Linguistic Segment Categories*, so that the scope of the **ana** attribute used to point to each interpretation is clearly defined. A further analysis into phrase and clause elements can be superimposed on the word and morpheme tagging in the preceding illustration. For example, CLAWS provides the following constituent analysis of the sample sentence (the word class codes have been deleted):

```
    [N [G The
    victim's G] friends N] [V told [N police N] [Fn that [N Krueger N] [V
    [V& drove [P into [N the quarry N]P]V&] and [V+ never surfaced
    V+]V]Fn]V]
```

Treating the labels on the brackets as phrase or clause interpretations, this analysis of the structure of the example sentence can be combined with the word class analysis and represented as follows (the symbol **V&** representing the first part of a coordinate phrase, has been replaced by **V1**, and **V+**, representing the second part, has been replaced by **V2**).

```
<s type="sentence">
  <phr ana="#n">
    <phr ana="#gn">
      <w ana="#AT0">The</w>
      <w ana="#NN1">victim</w>
      <m ana="#POS">'s</m>
    </phr>
    <w ana="#NN2">friends</w>
  </phr>
  <phr ana="#v">
    <w ana="#VVD">told</w>
    <phr ana="#n">
      <w ana="#NN2">police</w>
    </phr>
    <cl ana="#fn">
      <w ana="#CJT">that</w>
```

```
    <phr ana="#n">
      <w ana="#NP0">Krueger</w>
    </phr>
    <phr ana="#v">
      <phr ana="#v1">
        <w ana="#VVD">drove</w>
        <phr ana="#pr">
          <w ana="#PRP">into</w>
          <phr ana="#n">
            <w ana="#AT0">the</w>
            <w ana="#NN1">quarry</w>
          </phr>
        </phr>
      </phr>
      <w ana="#CJC">and</w>
      <phr ana="#v2">
        <w ana="#AV0">never</w>
        <w ana="#VVD">surfaced</w>
      </phr>
    </phr>
  </cl>
 </phr>
 <c ana="#pun">.</c>
</s>
```

This approach requires the definition of further <interp> (or <fs>) elements to provide targets for the pointers used to represent the constituent analysis:

```
<interpGrp type="constituentFunction">
 <interp xml:id="v2">coordinate continuation</interp>
 <interp xml:id="v">verbal</interp>
 <interp xml:id="n">nominal</interp>
 <interp xml:id="gn">genitive</interp>
 <interp xml:id="fn">finite clause</interp>
 <interp xml:id="pr">prepositional</interp>
 <interp xml:id="v1">coordinate start</interp>
</interpGrp>
```

Alternatively, a 'stand-off' representation for these analyses might be created using the <linkGrp> element. In this case, each linguistic segment must be supplied with its own **xml:id** attribute:

```
<s>
 <w xml:id="word-1">The</w>
 <w xml:id="word-2">victim</w>
 <w xml:id="word-3">'s</w>
 <w xml:id="word-4">friends</w>
 <w xml:id="word-5">told</w>
 <w xml:id="word-6">police</w>
 <w xml:id="word-7">that</w>
 <w xml:id="word-8">Kruger</w>
 <w xml:id="word-9">drove</w>
 <w xml:id="word10">into</w>
 <w xml:id="word11">the</w>
 <w xml:id="word12">quarry</w>
```

```
  <w  xml:id="word13">and</w>
  <w  xml:id="word14">never</w>
  <w  xml:id="word15">surfaced</w>
</s>
```

Each segment-interpretation pair may now be represented by means of a <link> element inside an appropriate <linkGrp> element:

```
<linkGrp  type="POS-annotation">
  <link  targets="#word-1 #AT0"/>
  <link  targets="#word-2 #NN1"/>
  <link  targets="#word-3 #POS"/>
  <link  targets="#word-4 #NN2"/>
  <link  targets="#word-5 #VVD"/>
  <link  targets="#word-6 #NN2"/>
<!--... -->
</linkGrp>
```

Each linguistic segment so far discussed has been well-behaved with respect to the basic document hierarchy, having only a single parent. Moreover, the segmentation has been complete, in that each part of the text is accounted for by some segment at each level of analysis, without discontinuities or overlap. This state of affairs does not of course apply in all types of analysis, and these Guidelines provide a number of mechanisms to support the representation of discontinuities or multiple analyses. A brief overview of these facilities is provided in chapter *20. Non-hierarchical Structures*; also see *16. Linking, Segmentation, and Alignment*. These mechanisms all depend to a greater or lesser degree on the use of pointing elements of various kinds.

The mechanisms proposed in this chapter may also be used to encode analyses of an entirely different kind, for example discourse function. Here is an application of the span technique to record details of a sales transaction in a spoken text.

```
<u  xml:id="u1">Can I have ten oranges and a kilo of bananas please?</u>
<u  xml:id="u2">Yes, anything else?</u>
<u  xml:id="u3">No thanks.</u>
<u  xml:id="u4">That'll be dollar forty.</u>
<u  xml:id="u5">Two dollars</u>
<u  xml:id="u6">Sixty, eighty, two dollars.  Thank you.</u>
<spanGrp  type="transactions">
  <span  from="#u1">sale request</span>
  <span  from="#u2"  to="#u3">sale compliance</span>
  <span  from="#u4">sale</span>
  <span  from="#u5">purchase</span>
  <span  from="#u6">purchase closure</span>
</spanGrp>
```

Source: [95]

For further discussion of the <u> (utterance) element and other elements recommended for transcriptions of spoken language, see chapter *8. Transcriptions of Speech*.

17.5 Module for Analysis and Interpretation

The module described in this chapter makes available the following components:

Module analysis: Simple analytic mechanisms

- Elements defined: c cl interp interpGrp m phr s span spanGrp w
- Classes defined: att.global.analytic

The selection and combination of modules to form a TEI schema is described in *1.2. Defining a TEI Schema*.

Chapter 18

Feature Structures

A feature structure is a general purpose data structure which identifies and groups together individual features, each of which associates a name with one or more values. Because of the generality of feature structures, they can be used to represent many different kinds of information, but they are of particular usefulness in the representation of linguistic analyses, especially where such analyses are partial, or underspecified. Feature structures represent the interrelations among various pieces of information, and their instantiation in markup provides a metalanguage for the generic representation of analyses and interpretations. Moreover, this instantiation allows feature values to be of specific types, and for restrictions to be placed on the values for particular features, by means of feature system declarations.[1]

18.1 Organization of this Chapter

This chapter is organized as follows. Following this introduction, section *18.2. Elementary Feature Structures and the Binary Feature Value* introduces the elements <fs> and <f>, used to represent feature structures and features respectively, together with the elementary binary feature value. Section *18.3. Other Atomic Feature Values* introduces elements for representing other kinds of atomic feature values such as symbolic, numeric, and string values. Section *18.4. Feature and Feature-Value Libraries* introduces the notion of predefined libraries or groups of features or feature values along with methods for referencing their components. Section *18.5. Feature Structures as Complex Feature Values* introduces complex values, in particular feature-structures as values, thus enabling feature structures to be recursively defined. Section *18.7. Collections as Complex Feature Values* discusses other complex values, in particular values which are collections, organized as sets, bags, and lists. Section *18.8. Feature Value Expressions* discusses how the operations of alternation, negation, and collection of feature values may be represented. Section *18.9. Default Values* discusses ways of representing underspecified, default, or uncertain values. Section *18.10. Linking Text and Analysis* discusses how analyses may be linked to other parts of an encoded text. Section *18.11. Feature System Declaration* describes the feature system declaration, a construct which provides for the validation of typed feature structures.

Formal definitions for all the elements introduced in this chapter are provided in section *18.12. Formal Definition and Implementation*.

18.2 Elementary Feature Structures and the Binary Feature Value

The fundamental elements used to represent a feature structure analysis are <f> (for feature), which represents a feature-value pair, and <fs> (for feature structure), which represents a structure made up of such feature-value pairs. The <fs> element has an optional **type** attribute which may be used to represent typed feature structures, and may contain any number of <f> elements. An <f> element has a required **name** attribute and an associated

[1] The recommendations of this chapter have been adopted as ISO Standard 24610-1 *Language Resource Management — Feature Structures — Part One: Feature Structure Representation*

value. The value may be simple: that is, a single binary, numeric, symbolic (i.e. taken from a restricted set of legal values), or string value, or a collection of such values, organized in various ways, for example, as a list; or it may be complex, that is, it may itself be a feature structure, thus providing a degree of recursion. Values may be under-specified or defaulted in various ways. These possibilities are all described in more detail in this and the following sections.

Feature and feature-value representations (including feature structure representations) may be embedded directly at any point in an XML document, or they may be collected together in special-purpose feature or feature-value libraries. The components of such libraries may then be referenced from other feature or feature-value representations, using the **feats** or **fVal** attribute as appropriate.

We begin by considering the simple case of a feature structure which contains binary-valued features only. The following three XML elements are needed to represent such a feature structure:

<fs> (feature structure) represents a feature structure, that is, a collection of feature-value pairs organized as a structural unit.

@**type** specifies the type of the feature structure.

@**feats** (features) references the feature-value specifications making up this feature structure.

<f> (feature) represents a feature value specification, that is, the association of a name with a value of any of several different types.

@**name** provides a name for the feature.

@**fVal** (feature value) references any element which can be used to represent the value of a feature.

<binary/> (binary value) represents the value part of a feature-value specification which can contain either of exactly two possible values.

The attributes **feats** and the **fVal** are not discussed in this section: they provide an alternative way of indicating the content of an element, as further discussed in section *18.4. Feature and Feature-Value Libraries*.

An <fs> element containing <f> elements with binary values can be straightforwardly used to encode the matrices of feature-value specifications for phonetic segments, such as the following for the English segment [s].

```
+--- ---+
| consonantal + |
| vocalic - |
| voiced - |
| anterior + |
| coronal + |
| continuant + |
| strident + |
+--- ---+
```

Source: [36]

This representation may be encoded in XML as follows:

```
<fs type="phonological_segments">
  <f name="consonantal">
    <binary value="true"/>
  </f>
  <f name="vocalic">
    <binary value="false"/>
  </f>
```

```
  <f name="voiced">
    <binary value="false"/>
  </f>
  <f name="anterior">
    <binary value="true"/>
  </f>
  <f name="coronal">
    <binary value="true"/>
  </f>
  <f name="continuant">
    <binary value="true"/>
  </f>
  <f name="strident">
    <binary value="true"/>
  </f>
</fs>
```

Note that <fs> elements may have an optional **type** attribute to indicate the kind of feature structure in question, whereas <f> elements must have a **name** attribute to indicate the name of the feature. Feature structures need not be typed, but features must be named. Similarly, the <fs> element may be empty, but the <f> element must have (or reference) some content.

The restriction of specific features to specific types of values (e.g. the restriction of the feature *strident* to a binary value) requires additional validation, as does any restriction on the features available within a feature structure of a particular type (e.g. whether a feature structure of type *phonological segment* necessarily contains a feature *voiced*). Such validation may be carried out at the document level, using special purpose processing, at the schema level using additional validation rules, or at the declarative level, using an additional mechanism such as the feature-system declaration discussed in *18.11. Feature System Declaration*.

Although we have used the term binary for this kind of value, and its representation in XML uses values such as `true` and `false` (or, equivalently, `1` and `0`), it should be noted that such values are not restricted to propositional assertions. As this example shows, this kind of value is intended for use with any binary-valued feature.

18.3 Other Atomic Feature Values

Features may take other kinds of atomic value. In this section, we define elements which may be used to represent: symbolic values, numeric values, and string values. The module defined by this chapter allows for the specification of additional datatypes if necessary, by extending the underlying class model.featureVal.single. If this is done, it is recommended that only the basic W3C datatypes should be used; more complex datatyping should be represented as feature structures.

<symbol/> (symbolic value) represents the value part of a feature-value specification which contains one of a finite list of symbols.

- @**value** supplies the symbolic value for the feature, one of a finite list that may be specified in a feature declaration.

<numeric/> (numeric value) represents the value part of a feature-value specification which contains a numeric value or range.

<string> (string value) represents the value part of a feature-value specification which contains a string.

The <symbol> element is used for the value of a feature when that feature can have any of a small, finite set of possible values, representable as character strings. For example, the following might be used to represent the claim that the Latin noun form *mensas* (tables) has accusative case, feminine gender, and plural number:

```
<fs>
  <f name="case">
    <symbol value="accusative"/>
  </f>
  <f name="gender">
    <symbol value="feminine"/>
  </f>
  <f name="number">
    <symbol value="plural"/>
  </f>
</fs>
```

More formally, this representation shows a structure in which three features (case, gender, and number) are used to define morpho-syntactic properties of a word. Each of these features can take one of a small number of values (for example, case can be `nominative`, `genitive`, `dative`, `accusative`, etc.) and it is therefore appropriate to represent the values taken in this instance as <symbol> elements. Note that, instead of using a symbolic value for grammatical number, one could have named the feature singular or plural and given it an appropriate binary value, as in the following example:

```
<fs>
  <f name="case">
    <symbol value="accusative"/>
  </f>
  <f name="gender">
    <symbol value="feminine"/>
  </f>
  <f name="singular">
    <binary value="false"/>
  </f>
</fs>
```

Whether one uses a binary or symbolic value in situations like this is largely a matter of taste.

The <string> element is used for the value of a feature when that value is a string drawn from a very large or potentially unbounded set of possible strings of characters, so that it would be impractical or impossible to use the <symbol> element. The string value is expressed as the content of the <string> element, rather than as an attribute value. For example, one might encode a street address as follows:

```
<fs>
  <f name="address">
    <string>3418 East Third Street</string>
  </f>
</fs>
```

The <numeric> element is used when the value of a feature is a numeric value, or a range of such values. For example, one might wish to regard the house number and the street name as different features, using an encoding like the following:

```
<fs>
  <f name="houseNumber">
    <numeric value="3418"/>
```

```
  </f>
  <f  name="streetName">
    <string>East Third Street</string>
  </f>
</fs>
```

If the numeric value to be represented falls within a specific range (for example an address that spans several numbers), the **max** attribute may be used to supply an upper limit:

```
<fs>
  <f  name="houseNumber">
    <numeric  value="3418"  max="3440"/>
  </f>
  <f  name="streetName">
    <string>East Third Street</string>
  </f>
</fs>
```

It is also possible to specify that the numeric value (or values) represented should (or should not) be truncated. For example, assuming that the daily rainfall in mm is a feature of interest for some address, one might represent this by an encoding like the following:

```
<fs>
  <f  name="dailyRainFall">
    <numeric  value="0.0"  max="1.3"  trunc="false"/>
  </f>
</fs>
```

This represents any of the infinite number of numeric values falling between 0 and 1.3; by contrast

```
<fs>
  <f  name="dailyRainFall">
    <numeric  value="0.0"  max="1.3"  trunc="true"/>
  </f>
</fs>
```

represents only two possible values: 0 and 1.

As noted above, additional processing is necessary to ensure that appropriate values are supplied for particular features, for example to ensure that the feature `singular` is not given a value such as **<symbol value="feminine"/>**. There are two ways of attempting to ensure that only certain combinations of feature names and values are used. First, if the total number of legal combinations is relatively small, one can predefine all of them in a construct known as a feature library, and then reference the combination required using the **feats** attribute in the enclosing <fs> element, rather than give it explicitly. This method is suitable in the situation described above, since it requires specifying a total of only ten (5 + 3 + 2) combinations of features and values. Similarly, to ensure that only feature structures containing valid combinations of feature values are used, one can put definitions for all valid feature structures inside a feature value library (so called, since a feature structure may be the value of a feature). A total of 30 feature structures (5 × 3 × 2) is required to enumerate all the possible combinations of individual case, gender and number values in the preceding illustration. We discuss the use of such libraries and their representation in XML further in section *18.4. Feature and Feature-Value Libraries* below.

However, the most general method of attempting to ensure that only legal combinations of feature names and values are used is to provide a feature-system declaration discussed in *18.11. Feature System Declaration*.

18.4 Feature and Feature-Value Libraries

As the examples in the preceding section suggest, the direct encoding of feature structures can be verbose. Moreover, it is often the case that particular feature-value combinations, or feature structures composed of them, are re-used in different analyses. To reduce the size and complexity of the task of encoding feature structures, one may use the **feats** attribute of the <fs> element to point to one or more of the feature-value specifications for that element. This indirect method of encoding feature structures presumes that the <f> elements are assigned unique **xml:id** values, and are collected together in <fLib> elements (feature libraries). In the same way, feature values of whatever type can be collected together in <fvLib> elements (feature-value libraries). If a feature has as its value a feature structure or other value which is predefined in this way, the **fVal** attribute may be used to point to it, as discussed in the next section. The following elements are used for representing feature, and feature-value libraries:

<fLib> (feature library) assembles a library of feature elements.

<fvLib> (feature-value library) assembles a library of reusable feature value elements (including complete feature structures).

For example, suppose a feature library for phonological feature specifications is set up as follows.

```
<fLib n="phonological features">
  <f xml:id="CNS1" name="consonantal">
    <binary value="true"/>
  </f>
  <f xml:id="CNS0" name="consonantal">
    <binary value="false"/>
  </f>
  <f xml:id="VOC1" name="vocalic">
    <binary value="true"/>
  </f>
  <f xml:id="VOC0" name="vocalic">
    <binary value="false"/>
  </f>
  <f xml:id="VOI1" name="voiced">
    <binary value="true"/>
  </f>
  <f xml:id="VOI0" name="voiced">
    <binary value="false"/>
  </f>
  <f xml:id="ANT1" name="anterior">
    <binary value="true"/>
  </f>
  <f xml:id="ANT0" name="anterior">
    <binary value="false"/>
  </f>
  <f xml:id="COR1" name="coronal">
    <binary value="true"/>
  </f>
  <f xml:id="COR0" name="coronal">
    <binary value="false"/>
  </f>
  <f xml:id="CNT1" name="continuant">
    <binary value="true"/>
  </f>
```

```
  <f  xml:id="CNT0"  name="continuant">
    <binary  value="false"/>
  </f>
  <f  xml:id="STR1"  name="strident">
    <binary  value="true"/>
  </f>
  <f  xml:id="STR0"  name="strident">
    <binary  value="false"/>
  </f>
<!-- ...  -->
</fLib>
```

Then the feature structures that represent the analysis of the phonological segments (phonemes) `/t/`, `/d/`, `/s/`, and `/z/` may be defined as follows.

```
<fs
   feats="#CNS1 #VOC0 #VOI0 #ANT1 #COR1 #CNT0 #STR0"/>
<fs
   feats="#CNS1 #VOC0 #VOI1 #ANT1 #COR1 #CNT0 #STR0"/>
<fs
   feats="#CNS1 #VOC0 #VOI0 #ANT1 #COR1 #CNT1 #STR1"/>
<fs
   feats="#CNS1 #VOC0 #VOI1 #ANT1 #COR1 #CNT1 #STR1"/>
```

The preceding are but four of the 128 logically possible fully specified phonological segments using the seven binary features listed in the feature library. Presumably not all combinations of features correspond to phonological segments (there are no strident vowels, for example). The legal combinations, however, can be collected together, each one represented as an identifiable <fs> element within a feature-value library, as in the following example:

```
<fvLib  xml:id="fsl1"  n="phonological segment definitions">
<!-- ...  -->
  <fs
     xml:id="T.DF"
     feats="#CNS1 #VOC0 #VOI0 #ANT1 #COR1 #CNT0 #STR0"/>
  <fs
     xml:id="D.DF"
     feats="#CNS1 #VOC0 #VOI1 #ANT1 #COR1 #CNT0 #STR0"/>
  <fs
     xml:id="S.DF"
     feats="#CNS1 #VOC0 #VOI0 #ANT1 #COR1 #CNT1 #STR1"/>
  <fs
     xml:id="Z.DF"
     feats="#CNS1 #VOC0 #VOI1 #ANT1 #COR1 #CNT1 #STR1"/>
<!-- ...  -->
</fvLib>
```

Once defined, these feature structure values can also be reused. Other <f> elements may invoke them by reference, using the **fVal** attribute; for example, one might use them in a feature value pair such as:

```
<f  name="dental-fricative"  fVal="#T.DF"/>
```

rather than expanding the hierarchy of the component phonological features explicitly.

Feature structures stored in this way may also be associated with the text which they are intended to annotate, either by a link from the text (for example, using the TEI global **ana** attribute), or by means of standoff annotation techniques (for example, using the TEI <link> element): see further section *18.10. Linking Text and Analysis* below.

Note that when features or feature structures are linked to in this way, the result is effectively a copy of the item linked to into the place from which it is linked. This form of linking should be distinguished from the phenomenon of structure-sharing, where it is desired to indicate that some part of an annotation structure appears simultaneously in two or more places within the structure. This kind of annotation should be represented using the <vLabel> element, as discussed in *18.6. Re-entrant Feature Structures* below.

18.5 Feature Structures as Complex Feature Values

Features may have complex values as well as atomic ones; the simplest such complex value is represented by supplying a <fs> element as the content of an <f> element, or (equivalently) by supplying the identifier of an <fs> element as the value for the **fVal** attribute on the <f> element. Structures may be nested as deeply as appropriate, using this mechanism. For example, an <fs> element may contain or point to an <f> element, which may contain or point to an <fs> element, which may contain or point to an <f> element, and so on.

To illustrate the use of complex values, consider the following simple model of a word, as a structure combining surface form information, a syntactic category, and semantic information. Each word analysis is represented as a **<fs type='word'>** element, containing three features named `surface`, `syntax`, and `semantics`. The first of these has an atomic string value, but the other two have complex values, represented as nested feature structures of types `category` and `act` respectively:

```
<fs type="word">
  <f name="surface">
    <string>love</string>
  </f>
  <f name="syntax">
    <fs type="category">
      <f name="pos">
        <symbol value="verb"/>
      </f>
      <f name="val">
        <symbol value="transitive"/>
      </f>
    </fs>
  </f>
  <f name="semantics">
    <fs type="act">
      <f name="rel">
        <symbol value="LOVE"/>
      </f>
    </fs>
  </f>
</fs>
```

This analysis does not tell us much about the meaning of the symbols `verb` or `transitive`. It might be preferable to replace these atomic feature values by feature structures. Suppose therefore that we maintain a feature-value library for each of the major syntactic categories (N, V, ADJ, PREP):

```
<fvLib n="Major category definitions">
<!-- ... -->
  <fs xml:id="N" type="noun">
<!-- noun features defined here -->
  </fs>
  <fs xml:id="V" type="verb">
<!-- verb features defined here -->
  </fs>
</fvLib>
```

This library allows us to use shortcut codes (`N`, `V`, etc.) to reference a complete definition for the corresponding feature structure. Each definition may be explicitly contained within the <fs> element, as a number of <f> elements. Alternatively, the relevant features may be referenced by their identifiers, supplied as the value of the **feats** attribute, as in these examples:

```
<!-- ... -->
<fs xml:id="ADJ" type="adjective" feats="#F1 #F2"/>
<fs xml:id="PREP" type="preposition" feats="#F1 #F3"/>
<!-- ... -->
```

This ability to re-use feature definitions within multiple feature structure definitions is an essential simplification in any realistic example. In this case, we assume the existence of a feature library containing specifications for the basic feature categories like the following:

```
<fLib n="categorial features">
  <f xml:id="NN-1" name="nominal">
    <binary value="true"/>
  </f>
  <f xml:id="NN-0" name="nominal">
    <binary value="false"/>
  </f>
  <f xml:id="VV-1" name="verbal">
    <binary value="true"/>
  </f>
  <f xml:id="VV-0" name="verbal">
    <binary value="false"/>
  </f>
<!-- ... -->
</fLib>
```

With such libraries in place, and assuming the availability of similarly predefined feature structures for transitivity and semantics, the preceding example could be considerably simplified:

```
<fs type="word">
  <f name="surface">
    <string>love</string>
  </f>
  <f name="syntax">
    <fs type="category">
      <f name="pos" fVal="#V"/>
      <f name="val" fVal="#TRNS"/>
```

```
    </fs>
  </f>
  <f  name="semantics">
    <fs  type="act">
      <f  name="rel"  fVal="#LOVE"/>
    </fs>
  </f>
</fs>
```

Although in principle the **fVal** attribute could point to any kind of feature value, its use is not recommended for simple atomic values.

18.6 Re-entrant Feature Structures

Sometimes the same feature value is required at multiple places within a feature structure, in particular where the value is only partially specified at one or more places. The <vLabel> element is provided as a means of labelling each such re-entrancy point:

<vLabel> (value label) represents the value part of a feature-value specification which appears at more than one point in a feature structure.

For example, suppose one wishes to represent noun-verb agreement as a single feature structure. Within the representation, the feature indicating (say) number appears more than once. To represent the fact that each occurrence is another appearance of the same feature (rather than a copy) one could use an encoding like the following:

```
<fs  xml:id="NVA">
  <f  name="nominal">
    <fs>
      <f  name="nm-num">
        <vLabel  name="L1">
          <symbol  value="singular"/>
        </vLabel>
      </f>
<!-- other nominal features -->
    </fs>
  </f>
  <f  name="verbal">
    <fs>
      <f  name="vb-num">
        <vLabel  name="L1"/>
      </f>
    </fs>
<!-- other verbal features -->
  </f>
</fs>
```

In the above encoding, the features named `vb-num` and `nm-num` exhibit structure sharing. Their values, given as `vLabel` elements, are understood to be references to the same point in the feature structure, which is labelled by their **name** attribute.

The scope of the names used to label re-entrancy points is that of the outermost <fs> element in which they appear. When a feature structure is imported from a feature value library, or referenced from elsewhere (for example by using the **fVal** attribute) the names of any sharing points it may contain are implicitly prefixed by the identifier used for the imported feature structure, to avoid name clashes. Thus, if some other feature structure were to reference the <fs> element given in the example above, for example in this way:

```
<f  name="class"  fVal="#NVA"/>
```

then the labelled points in the example would be interpreted as if they had the name NVAL1.

18.7 Collections as Complex Feature Values

Complex feature values need not always be represented as feature structures. Multiple values may also be organized as sets, bags or multisets, or lists of atomic values of any type. The <vColl> element is provided to represent such cases:

<vColl> (collection of values) represents the value part of a feature-value specification which contains multiple values organized as a set, bag, or list.

A feature whose value is regarded as a set, bag, or list may have any positive number of values as its content, or none at all, (thus allowing for representation of the empty set, bag, or list). The items in a list are ordered, and need not be distinct. The items in a set are not ordered, and must be distinct. The items in a bag are neither ordered nor distinct. Sets and bags are thus distinguished from lists in that the order in which the values are specified does not matter for the former, but does matter for the latter, while sets are distinguished from bags and lists in that repetitions of values do not count for the former but do count for the latter.

If no value is specified for the **org** attribute, the assumption is that the <vColl> defines a list of values. If the <vColl> element is empty, the assumption is that it represents the null list, set, or bag.

To illustrate the use of the **org** attribute, suppose that a feature structure analysis is used to represent a genealogical tree, with the information about each individual treated as a single feature structure, like this:

```
<fs  xml:id="p027"  type="person">
  <f  name="forenames">
    <vColl>
      <string>Daniel</string>
      <string>Edouard</string>
    </vColl>
  </f>
  <f  name="mother"  fVal="#p002"/>
  <f  name="father"  fVal="#p009"/>
  <f  name="birthDate">
    <fs  type="date"  feats="#y1988 #m04 #d17"/>
  </f>
  <f  name="birthPlace"  fVal="#austintx"/>
  <f  name="siblings">
    <vColl  org="set">
      <fs  copyOf="#pnb005"/>
      <fs  copyOf="#prb001"/>
    </vColl>
  </f>
</fs>
```

In this example, the <vColl> element is first used to supply a list of 'name' feature values, which together constitute the 'forenames' feature. Other features are defined by reference to values which we assume are held in some external feature value library (not shown here). For example, the <vColl> element is used a second time to indicate that the persons's siblings should be regarded as constituting a set rather than a list. Each sibling is represented by a feature structure: in this example, each feature structure is a copy of one specified in the feature value library.

If a specific feature contains only a single feature structure as its value, the component features of which are organized as a set, bag, or list, it may be more convenient to represent the value as a <vColl> rather than

as a <fs>. For example, consider the following encoding of the English verb form *sinks*, which contains an *agreement* feature whose value is a feature structure which contains *person* and *number* features with symbolic values.

```
<fs type="word">
  <f name="category">
    <symbol value="verb"/>
  </f>
  <f name="tense">
    <symbol value="present"/>
  </f>
  <f name="agreement">
    <fs>
      <f name="person">
        <symbol value="third"/>
      </f>
      <f name="number">
        <symbol value="singular"/>
      </f>
    </fs>
  </f>
</fs>
```

If the names of the features contained within the *agreement* feature structure are of no particular significance, the following simpler representation may be used:

```
<fs type="word">
  <f name="category">
    <symbol value="verb"/>
  </f>
  <f name="tense">
    <symbol value="present"/>
  </f>
  <f name="agreement">
    <vColl org="set">
      <symbol value="third"/>
      <symbol value="singular"/>
    </vColl>
  </f>
</fs>
```

The <vColl> element is also useful in cases where an analysis has several components. In the following example, the French word *auxquels* has a two-part analysis, represented as a list of two values. The first specifies that the word contains a preposition; the second that it contains a masculine plural relative pronoun:

```
<fs>
  <f name="lex">
    <symbol value="auxquels"/>
  </f>
  <f name="maf">
    <vColl org="list">
      <fs>
        <f name="cat">
          <symbol value="prep"/>
```

```
      </f>
     </fs>
     <fs>
      <f name="cat">
       <symbol value="pronoun"/>
      </f>
      <f name="kind">
       <symbol value="rel"/>
      </f>
      <f name="num">
       <symbol value="pl"/>
      </f>
      <f name="gender">
       <symbol value="masc"/>
      </f>
     </fs>
    </vColl>
  </f>
</fs>
```

The set, bag, or list which has no members is known as the null (or empty) set, bag, or list. A <vColl> element with no content and with no value for its **feats** attribute is interpreted as referring to the null set, bag, or list, depending on the value of its **org** attribute.

If, for example, the individual described by the feature structure with identifier `p027` (above) had no siblings, we might specify the *siblings* feature as follows.

```
<f name="siblings">
  <vColl org="set"/>
</f>
```

A <vColl> element may also collect together one or more other <vColl> elements, if, for example one of the members of a set is itself a set, or if two lists are concatenated together. Note that such collections pay no attention to the contents of the nested <vColl> elements: if it is desired to produce the union of two sets, the <vMerge> element discussed below should be used to make a new collection from the two sets.

18.8 Feature Value Expressions

It is sometimes desirable to express the value of a feature as the result of an operation over some other value (for example, as 'not green', or as 'male or female', or as the concatenation of two collections). Three special purpose elements are provided to represent disjunctive alternation, negation, and collection of values:

<vAlt> (value alternation) represents the value part of a feature-value specification which contains a set of values, only one of which can be valid.

<vNot> (value negation) represents a feature value which is the negation of its content.

<vMerge> (merged collection of values) represents a feature value which is the result of merging together the feature values contained by its children, using the organization specified by the **org** attribute.

18.8.1 Alternation

The <vAlt> element can be used wherever a feature value can appear. It contains two or more feature values, any one of which is to be understood as the value required. Suppose, for example, that we are using a feature system to describe residential property, using such features as *number.of.bathrooms*. In a particular case, we

might wish to represent uncertainty as to whether a house has two or three bathrooms. As we have already shown, one simple way to represent this would be with a numeric maximum:

```
<f name="number.of.bathrooms">
  <numeric value="2" max="3"/>
</f>
```

A more general way would be to represent the alternation explicitly, in this way:

```
<f name="number.of.bathrooms">
  <vAlt>
    <numeric value="2"/>
    <numeric value="3"/>
  </vAlt>
</f>
```

The <vAlt> element represents alternation over feature values, not feature-value pairs. If therefore the uncertainty relates to two or more feature value specifications, each must be represented as a feature structure, since a feature structure can always appear where a value is required. For example, suppose that it is uncertain as to whether the house being described has two bathrooms or two bedrooms, a structure like the following may be used:

```
<f name="rooms">
  <vAlt>
    <fs>
      <f name="number.of.bathrooms">
        <numeric value="2"/>
      </f>
    </fs>
    <fs>
      <f name="number.of.bedrooms">
        <numeric value="2"/>
      </f>
    </fs>
  </vAlt>
</f>
```

Note that alternation is always regarded as exclusive: in the case above, the implication is that having two bathrooms excludes the possibility of having two bedrooms and vice versa. If inclusive alternation is required, a <vColl> element may be included in the alternation as follows:

```
<f name="rooms">
  <vAlt>
    <fs>
      <f name="number.of.bathrooms">
        <numeric value="2"/>
      </f>
    </fs>
    <fs>
      <f name="number.of.bedrooms">
        <numeric value="2"/>
      </f>
```

```
    </fs>
    <vColl>
      <fs>
        <f  name="number.of.bathrooms">
          <numeric  value="2"/>
        </f>
      </fs>
      <fs>
        <f  name="number.of.bedrooms">
          <numeric  value="2"/>
        </f>
      </fs>
    </vColl>
  </vAlt>
</f>
```

This analysis indicates that the property may have two bathrooms, two bedrooms, or both two bathrooms and two bedrooms.

As the previous example shows, the <vAlt> element can also be used to indicate alternations among values of features organized as sets, bags or lists. Suppose we use a feature `selling.points` to describe items that are mentioned to enhance a property's sales value, such as whether it has a pool or a good view. Now suppose for a particular listing, the selling points include an alarm system and a good view, and either a pool or a jacuzzi (but not both). This situation could be represented, using the <vAlt> element, as follows.

```
<fs  type="real_estate_listing">
  <f  name="selling.points">
    <vColl  org="set">
      <string>alarm system</string>
      <string>good view</string>
      <vAlt>
        <string>pool</string>
        <string>jacuzzi</string>
      </vAlt>
    </vColl>
  </f>
</fs>
```

Now suppose the situation is like the preceding except that one is also uncertain whether the property has an alarm system or a good view. This can be represented as follows.

```
<fs  type="real_estate_listing">
  <f  name="selling.points">
    <vColl  org="set">
      <vAlt>
        <string>alarm system</string>
        <string>good view</string>
      </vAlt>
      <vAlt>
        <string>pool</string>
        <string>jacuzzi</string>
      </vAlt>
    </vColl>
  </f>
</fs>
```

If a large number of ambiguities or uncertainties need to be represented, involving a relatively small number of features and values, it is recommended that a stand-off technique, for example using the general-purpose <alt> element discussed in section *16.8. Alternation* be used, rather than the special-purpose <vAlt> element.

18.8.2 Negation

The <vNot> element can be used wherever a feature value can appear. It contains any feature value and returns the complement of its contents. For example, the feature *number.of.bathrooms* in the following example has any whole numeric value other than 2:

```
<f  name="number.of.bathrooms">
  <vNot>
    <numeric  value="2"/>
  </vNot>
</f>
```

Strictly speaking, the effect of the <vNot> element is to provide the complement of the feature values it contains, rather than their negation. If a feature system declaration is available which defines the possible values for the associated feature, then it is possible to say more about the negated value. For example, suppose that the available values for the feature `case` are declared to be nominative, genitive, dative, or accusative, whether in a TEI feature system declaration or by some other means. Then the following two specifications are equivalent:

```
 (i) <f  name="case">
  <vNot>
    <symbol  value="genitive"/>
  </vNot>
</f>
(ii)
<f  name="case">
  <vAlt>
    <symbol  value="nominative"/>
    <symbol  value="dative"/>
    <symbol  value="accusative"/>
  </vAlt>
</f>
```

If however no such system declaration is available, all that one can say about a feature specified via negation is that its value is something other than the negated value.

Negation is always applied to a feature value, rather than to a feature-value pair. The negation of an atomic value is the set of all other values which are possible for the feature.

Any kind of value can be negated, including collections (represented by a <vColl> elements) or feature structures (represented by <fs> elements). The negation of any complex value is understood to be the set of values which cannot be unified with it. Thus, for example, the negation of the feature structure F is understood to be the set of feature structures which are not unifiable with F. In the absence of a constraint mechanism such as the Feature System Declaration, the negation of a collection is anything that is not unifiable with it, including collections of different types and atomic values. It will generally be more useful to require that the organization of the negated value be the same as that of the original value, for example that a negated set is understood to mean the set which is a complement of the set, but such a requirement cannot be enforced in the absence of a constraint mechanism.

18.8.3 Collection of Values

The <vMerge> element can be used wherever a feature value can appear. It contains two or more feature values, all of which are to be collected together. The organization of the resulting collection is specified by the value of the **org** attribute, which need not necessarily be the same as that of its constituent values if these are collections. For example, one can change a list to a set, or vice versa.

As an example, suppose that we wish to represent the range of possible values for a feature 'genders' used to describe some language. It would be natural to represent the possible values as a set, using the <vColl> element as in the following example:

```
<fs>
  <f  name="genders">
    <vColl  org="set">
      <symbol  value="masculine"/>
      <symbol  value="feminine"/>
    </vColl>
  </f>
</fs>
```

Suppose however that we discover for some language it is necessary to add a new possible value, and to treat the value of the feature as a list rather than as a set. The <vMerge> element can be used to achieve this:

```
<fs>
  <f  name="genders">
    <vMerge  org="list">
      <vColl  org="set">
        <symbol  value="masculine"/>
        <symbol  value="feminine"/>
      </vColl>
      <symbol  value="neuter"/>
    </vMerge>
  </f>
</fs>
```

18.9 Default Values

The value of a feature may be underspecified in a number of different ways. It may be null, unknown, or uncertain with respect to a range of known possibilities, as well as being defined as a negation or an alternation. As previously noted, the specification of the range of known possibilities for a given feature is not part of the current specification: in the TEI scheme, this information is conveyed by the feature system declaration. Using this, or some other system, we might specify (for example) that the range of values for an element includes symbols for masculine, feminine, and neuter, and that the default value is neuter. With such definitions available to us, it becomes possible to say that some feature takes the default value, or some unspecified value from the list. The following special element is provided for this purpose:

<default/> (default feature value) represents the value part of a feature-value specification which contains a defaulted value.

The value of an empty <f> element which also lacks a **fVal** attribute is understood to be the most general case, i.e. any of the available values. Thus, assuming the feature system defined above, the following two representations are equivalent.

```
<f  name="gender"/>
<f  name="gender">
  <vAlt>
    <symbol  value="feminine"/>
    <symbol  value="masculine"/>
    <symbol  value="neuter"/>
  </vAlt>
</f>
```

If, however, the value is explicitly stated to be the default one, using the <default> element, then the following two representations are equivalent:

```
<f  name="gender">
  <default/>
</f>
```

```
<f  name="gender">
  <symbol  value="neuter"/>
</f>
```

Similarly, if the value is stated to be the negation of the default, then the following two representations are equivalent:

```
<f  name="gender">
  <vNot>
    <default/>
  </vNot>
</f>
```

```
<f  name="gender">
  <vAlt>
    <symbol  value="feminine"/>
    <symbol  value="masculine"/>
  </vAlt>
</f>
```

18.10 Linking Text and Analysis

Text elements can be linked with feature structures using any of the linking methods discussed elsewhere in the Guidelines (see for example sections *17.2. Global Attributes for Simple Analyses* and *17.4. Linguistic Annotation*). In the simplest case, the **ana** attribute may be used to point from any element to an annotation of it, as in the following example:

```
<s  n="00741">
  <w  ana="#at0">The</w>
  <w  ana="#ajs">closest</w>
  <w  ana="#pnp">he</w>
  <w  ana="#vvd">came</w>
```

```
  <w  ana="#prp">to</w>
  <w  ana="#nn1">exercise</w>
  <w  ana="#vbd">was</w>
  <w  ana="#to0">to</w>
  <w  ana="#vvi">open</w>
  <w  ana="#crd">one</w>
  <w  ana="#nn1">eye</w>
  <phr  ana="#av0">
    <w>every</w>
    <w>so</w>
    <w>often</w>
  </phr>
  <c  ana="#pun">,</c>
  <w  ana="#cjs">if</w>
  <w  ana="#pni">someone</w>
  <w  ana="#vvd">entered</w>
  <w  ana="#at0">the</w>
  <w  ana="#nn1">room</w>
<!-- ... -->
</s>
```

The values specified for the **ana** attribute reference components of a feature-structure library, which represents all of the grammatical structures used by this encoding scheme. (For illustrative purposes, we cite here only the structures needed for the first six words of the sample sentence):

```
<fvLib  xml:id="C6"  n="Claws 6 tags">
<!-- ... -->
  <fs  xml:id="ajs"  type="grammatical_structure"  feats="#wj #ds"/>
  <fs  xml:id="at0"  type="grammatical_structure"  feats="#wl"/>
  <fs  xml:id="pnp"  type="grammatical_structure"  feats="#wr #rp"/>
  <fs  xml:id="vvd"  type="grammatical_structure"  feats="#wv #bv #fd"/>
  <fs  xml:id="prp"  type="grammatical_structure"  feats="#wp #bp"/>
  <fs  xml:id="nnn"  type="grammatical_structure"  feats="#wn #tc #ns"/>
<!-- ... -->
</fvLib>
```

The components of each feature structure in the library are referenced in much the same way, using the **feats** attribute to identify one or more <f> elements in the following feature library (again, only a few of the available features are quoted here):

```
<fLib>
<!-- ... -->
  <f  xml:id="bv"  name="verbbase">
    <symbol  value="main"/>
  </f>
  <f  xml:id="bp"  name="prepbase">
    <symbol  value="lexical"/>
  </f>
  <f  xml:id="ds"  name="degree">
    <symbol  value="superlative"/>
  </f>
  <f  xml:id="fd"  name="verbform">
    <symbol  value="ed"/>
  </f>
```

```
  <f  xml:id="ns"  name="number">
    <symbol  value="singular"/>
  </f>
  <f  xml:id="rp"  name="prontype">
    <symbol  value="personal"/>
  </f>
  <f  xml:id="tc"  name="nountype">
    <symbol  value="common"/>
  </f>
  <f  xml:id="wj"  name="class">
    <symbol  value="adjective"/>
  </f>
  <f  xml:id="wl"  name="class">
    <symbol  value="article"/>
  </f>
  <f  xml:id="wn"  name="class">
    <symbol  value="noun"/>
  </f>
  <f  xml:id="wp"  name="class">
    <symbol  value="preposition"/>
  </f>
  <f  xml:id="wr"  name="class">
    <symbol  value="pronoun"/>
  </f>
  <f  xml:id="wv"  name="class">
    <symbol  value="verb"/>
  </f>
<!-- ...  -->
</fLib>
```

Alternatively, a stand-off technique may be used, as in the following example, where a <linkGrp> element is used to link selected characters in the text *Caesar seized control* with their phonological representations.

```
<s>
  <w  xml:id="S1W1">
    <c  xml:id="S1W1C1">C</c>ae<c  xml:id="S1W1C2">s</c>ar</w>
  <w  xml:id="S1W2">
    <c  xml:id="S1W2C1">s</c>ei<c  xml:id="S1W2C2">z</c>e<c  xml:id="S1W2C3">d</c>
  </w>
  <w  xml:id="S1W3">con<c  xml:id="S1W3C1">t</c>rol</w>.

</s>
<fvLib  xml:id="FSL1"  n="phonological segment definitions">
<!-- as in previous example -->
</fvLib>
<linkGrp  type="phonology">
<!-- ...  -->
  <link  targets="#S.DF #S1W3C1"/>
  <link  targets="#Z.DF #S1W2C3"/>
  <link  targets="#S.DF #S1W2C1"/>
  <link  targets="#Z.DF #S1W2C2"/>
<!-- ...  -->
</linkGrp>
```

As this example shows, a stand-off solution requires that every component to be linked to must be

addressable in some way, by means of an XPointer. To handle the POS tagging example above, for example, each annotated element might be given an identifier of some sort, as follows:

```
<s  xml:id="mds09"  n="00741">
  <w  xml:id="mds0901">The</w>
  <w  xml:id="mds0902">closest</w>
  <w  xml:id="mds0903">he</w>
  <w  xml:id="mds0904">came</w>
  <w  xml:id="mds0905">to</w>
  <w  xml:id="mds0906">exercise</w>
<!-- ... -->
</s>
```

It would then be possible to link each word to its intended annotation in the feature library quoted above, as follows:

```
<linkGrp  type="POS-codes">
<!-- ... -->
  <link  targets="#mds0901 #at0"/>
  <link  targets="#mds0902 #ajs"/>
  <link  targets="#mds0903 #pnp"/>
  <link  targets="#mds0904 #vvd"/>
  <link  targets="#mds0905 #prp"/>
  <link  targets="#mds0906 #nn1"/>
  <link  targets="#mds0907 #vbd"/>
  <link  targets="#mds0908 #to0"/>
  <link  targets="#mds0909 #vvi"/>
  <link  targets="#mds0910 #crd"/>
<!-- ... -->
</linkGrp>
```

18.11 Feature System Declaration

The Feature System Declaration (FSD) is intended for use in conjunction with a TEI-conforming text that makes use of <fs> (that is, feature structure) elements. The FSD serves three purposes:

- It provides a mechanism by which the encoder can list all of the feature names and feature values and give a prose description as to what each represents.
- It provides a mechanism by which the encoder can define constraints not only what it means to be a well-formed feature structure, but also **valid** feature structure, relative to a given theory stated in typed feature logic. These constraints may involve constraints on the range of a feature value, constraints on what features are valid within certain types of feature structures, or constraints that prevent the co-occurrence of certain feature-value pairs.
- It provides a mechanism by which the encoder can define the intended interpretation of underspecified feature structures. This involves defining default values (whether literal or computed) for missing features.

The scheme described in this chapter may be used to document any feature structure system, but is primarily intended for use with the feature structure representation defined by the ISO 24610-1:2006 standard, which corresponds with the recommendations presented in these Guidelines, *18. Feature Structures*. This chapter relies upon, but does not reproduce, formal definitions and descriptions presented more thoroughly in the ISO standard, which should be consulted in case of ambiguity or uncertainty.

The FSD serves an important function in documenting precisely what the encoder intended by the system of feature structure markup used in an XML-encoded text. The FSD is also an important resource which

standardizes the rules of inference used by software to validate the feature structure markup in a text, and to infer the full interpretation of underspecified feature structures.

The reader should be aware the terminology used in this document does not always closely follow conventional practice in formal logic, and may also diverge from practice in some linguistic applications of typed feature structures. In particular, the term 'interpretation' when applied to a feature structure is not an interpretation in the model-theoretic sense, but is instead a minimally informative (or equivalently, most general) extension of that feature structure that is consistent with a set of constraints declared by an FSD. In linguistic application, such a system of constraints is the principal means by which the grammar of some natural language is expressed. There is a great deal of disagreement as to what, if any, model-theoretic interpretation feature structures have in such applications, but the status of this formal kind of interpretation is not germane to the present document. Similarly, the term 'valid' is used here as elsewhere in these Guidelines to identify the syntactic state of well-formedness in the sense defined by the logic of typed feature structures itself, as distinct from and in addition to the 'well-formedness' that pertains at the level of this encoding standard. No appeal to any notion from formal semantics should be inferred.

We begin by describing how an encoded text is associated with one or more feature system declarations. The second, third, and fourth sections describe the overall structure of a feature system declaration and give details of how to encode its components. The final section offers a full example; fuller discussion of the reasoning behind FSDs and another complete example are provided in Langendoen and Simons (1995).

18.11.1 Linking a TEI Text to Feature System Declarations

In order for application software to use feature system declarations to aid in the automatic interpretation of encoded texts, or even for human readers to find the appropriate declarations which document the feature system used in markup, there must be a formal link from the encoded texts to the declarations. However, the schema which declares the syntax of the Feature System itself should be kept distinct from the feature structure schema, which is an application of that system.

A document containing typed feature structures may simply include a feature system declaration documenting those feature structures. A more usual scenario, however, is that the same feature system declaration (or parts of it) will be shared by many documents. In either case, an <fsDecl> element for each distinct type of feature structure used must be provided and associated with the type, which is the value used within each feature structure for its **type** attribute.

When the module defined in this chapter is included in an XML schema, the following elements become available:

<fsdDecl> (feature system declaration) provides a feature system declaration comprising one or more feature structure declarations or feature structure declaration links.

<fsdLink/> (feature structure declaration link) associates the name of a typed feature structure with a feature structure declaration for it.

<fsDecl> (feature structure declaration) declares one type of feature structure.

The <fsdDecl> element may be supplied either within the header of a standard TEI document, or as a standalone document in its own right. It contains one or more <fsdLink> or <fsDecl> elements.

For example, suppose that a document doc.xml contains feature structures of two types: **gpsg** and **lex**. We might simply embed an <fsDecl> element for each within the header attached to the document as follows:

```
<TEI>
  <teiHeader>
    <fileDesc>
<!-- doc1 -->
    </fileDesc>
```

```
   <encodingDesc>
<!-- ... -->
     <fsdDecl>
       <fsDecl type="gpsg">
<!-- information about this type -->
       </fsDecl>
       <fsDecl type="lex">
<!-- information about this type -->
       </fsDecl>
     </fsdDecl>
<!-- ... -->
   </encodingDesc>
  </teiHeader>
  <text>
    <body>
      <fs type="lex">
<!-- an instance of the typed feature structure "lex" -->
     </fs>
   </body>
  </text>
</TEI>
```

In this case there is an implicit link between the <fs> element and the corresponding <fsDecl> element because they share the same value for their **type** attribute and appear within the same document. This is a short cut for the more general case which requires a more explicit link provided by means of the <fsdLink> element, as demonstrated below.

Now suppose that we wish to create a second document which includes feature structures of the same type. Rather than duplicate the corresponding declarations, we will need to provide a means of pointing to them from this second document. The easiest[2] way of accomplishing this is to add an XML identifier to each <fsDecl> element in doc1.xml:

```
<!-- ... --><fsdDecl>
  <fsDecl type="gpsg" xml:id="GPSG">
<!-- information about this type -->
  </fsDecl>
  <fsDecl type="lex" xml:id="LEX">
<!-- information about this type -->
  </fsDecl>
</fsdDecl>
```

(Although in this case the XML identifier is simply an uppercase version of the type name, there is no necessary connection between the two names. The only requirement is that the XML identifier conform to the standards required for identifiers, and that it be unique within the document containing it.)

In the <fsdDecl> for the second document, we can now include pointers to the <fsDecl> elements in the first:

```
<TEI>
  <teiHeader>
    <fileDesc>
```

[2] Ways of pointing to components of a TEI document without using an XML identifier are discussed in *16.2.1. Pointing Elsewhere*

```
<!-- doc2 -->
   </fileDesc>
   <encodingDesc>
<!-- ... -->
     <fsdDecl>
       <fsdLink  type="gpsg"  target="doc1.xml#GPSG"/>
       <fsdLink  type="lexx"  target="doc1.xml#GPSG"/>
     </fsdDecl>
<!-- ... -->
   </encodingDesc>
  </teiHeader>
  <text>
    <body>
      <fs  type="lexx">
<!-- an instance of the typed feature structure "lex" -->
     </fs>
   </body>
  </text>
</TEI>
```

Note that in doc2.xml there is no requirement for the local name for a given type of feature structures to be the same as that used by doc1.xml. We assume in this encoding that the type called lexx in doc2.xml is declared as having identical constraints and other properties to those declared for the type called lex in doc1.xml.

A <fsdDecl> may be given, as above, within the encoding description of the <teiHeader> element of a TEI document containing typed feature structures. Alternatively, it may appear independently of any feature structures, as a document in its own right, possibly with its own <teiHeader>. These options are both possible because the element is a member of both the model.encodingPart class and the model.resourceLike class.

The current recommendations provide no way of enforcing uniqueness of the **type** values among <fsdDecl> elements, nor of requiring that every **type** value specified on a <fs> element be also declared on an <fsdDecl> element. Encoders requiring such constraints (which might have some obvious utility in assisting the consistency and accuracy of tagging) are recommended to develop tools to enforce them, using such mechanisms as Schematron assertions.

18.11.2 The Overall Structure of a Feature System Declaration

A feature system declaration contains one or more feature structure declarations, each of which has up to three parts: an optional description (which gives a prose comment on what that type of feature structure encodes), an obligatory set of feature declarations (which specify range constraints and default values for the features in that type of structure), and optional feature structure constraints (which specify co-occurrence restrictions on feature values).

<fsDescr> (feature system description (in FSD)) describes in prose what is represented by the type of feature structure declared in the enclosing fsDecl.

<fDecl> (feature declaration) declares a single feature, specifying its name, organization, range of allowed values, and optionally its default value.

<fsConstraints> (feature-structure constraints) specifies constraints on the content of valid feature structures.

Feature declarations and feature structure constraints are described in the next two sections. Note that the specification of similar <fsDecl> elements can be simplified by devising an inheritance hierarchy for the feature structure types. Each <fsDecl> element may name one or more 'basetypes' from which it inherits feature declarations and constraints (these are often called 'supertypes'). For instance, suppose that **<fsDecl**

type="Basic"> contains <**fDecl name="One"**> and <**fDecl name="Two"**>, and that <**fsDecl type="Derived" baseTypes="Basic"**> contains just <**fDecl name="Three"**>. Then any instance of <**fs type="Derived"**> must include all three features. This is because <**fsDecl type="Derived"**> inherits the two feature declarations from <**fsDecl type="Basic"**> when it specifies a base type of **Basic**.

The following sample shows the overall structure of a complete feature structure declaration:

```
<fsDecl  type="SomeName">
  <fsDescr>Describes what this type of fs represents</fsDescr>
  <fDecl  name="featureOne">
<!-- The declaration for featureOne -->
  </fDecl>
  <fDecl  name="featureTwo">
<!-- The declaration for featureTwo -->
  </fDecl>
  <fsConstraints>
<!-- The feature structure constraints go here -->
  </fsConstraints>
</fsDecl>
```

The attribute **baseTypes** gives the name of one or more types from which this type inherits feature specifications and constraints; if this type includes a feature specification with the same name as one inherited from any of the types specified by this attribute, or if more than one specification of the same name is inherited, then the possible values of that feature is determined by unification. Similarly, the set of constraints applicable is derived by conjoining those specified explicitly within this element with those implied by the **baseTypes** attribute. When no base type is specified, no feature specification or constraint is inherited.

Although the present standard does provide for default feature values, feature inheritance is defined to be monotonic.

The process of combining constraints may result in a contradiction, for example if two specifications for the same feature specify disjoint ranges of values, and at least one such specification is mandatory. In such a case, there is no valid feature structure of the type being defined.

Every type specified by **baseTypes** must be a single word which is a legal XML name; for example, they cannot include whitespace or begin with digits. Multiple base types are separated with spaces, e.g. <**fsDecl type="Sub" baseTypes="Super1 Super2"**>.

18.11.3 Feature Declarations

Each feature is declared in an <fDecl> element whose **name** attribute identifies the feature being declared; this matches the **name** attribute of the <f> elements it declares.

An <fDecl> has three parts: an optional prose description (which should explain what the feature and its values represent), an obligatory range specification (which declares what values the feature is allowed to have), and an optional default specification (which declares what default value should be supplied when the named feature does not appear in an <fs>). If, in a feature structure, a feature:

- is not optional (i.e., is obligatory),
- has no value provided, or the value <default> is provided (see ISO 24610-1, Subclause 5.10, Default Values, and
- either has no default specified, or has conditional defaults, none of the conditions on which is met,

then the value of this feature in the feature structure's most general valid extension is the most general value provided in its <vRange>, in the case of a unit organization, or the singleton set, bag, or list containing that element, in the case of a complex organization. If the feature:

- is optional,
- has no value provided, or the value <default> is provided, and
- either has a default specified, or has conditional defaults, one of the conditions on which is met,

then this feature does have a value in the feature structure's most general valid extension when it exists, namely the default value that pertains.

It is possible that a feature structure will not have a valid extension because the default value that pertains to a feature is not consistent with that feature's declared range. Additional tools are required for the enforcement of such criteria.

The following elements are used in feature system declarations:

<fDecl> (feature declaration) declares a single feature, specifying its name, organization, range of allowed values, and optionally its default value.

@**name** indicates the name of the feature being declared; matches the **name** attribute of <f> elements in the text.

@**optional** indicates whether or not the value of this feature may be present.

<fDescr> (feature description (in FSD)) describes in prose what is represented by the feature being declared and its values.

<vRange> (value range) defines the range of allowed values for a feature, in the form of an <fs>, <vAlt>, or primitive value; for the value of an <f> to be valid, it must be subsumed by the specified range; if the <f> contains multiple values (as sanctioned by the **org** attribute), then each value must be subsumed by the <vRange>.

<vDefault> (value default) declares the default value to be supplied when a feature structure does not contain an instance of <f> for this name; if unconditional, it is specified as one (or, depending on the value of the **org** attribute of the enclosing <fDecl>) more <fs> elements or primitive values; if conditional, it is specified as one or more <if> elements; if no default is specified, or no condition matches, the value **none** is assumed.

<if> defines a conditional default value for a feature; the condition is specified as a feature structure, and is met if it subsumes the feature structure in the text for which a default value is sought.

<then/> separates the condition from the default in an <if>, or the antecedent and the consequent in a <cond> element.

The logic for validating feature values and for matching the conditions for supplying default values is based on the operation of subsumption. Subsumption is a standard operation in feature-structure-based formalisms. Informally, a feature structure FS subsumes all feature structures that are at least as informative as itself; that is, all feature structures that specify all of the feature values that FS does with values that are subsumed by the values that FS has, and that have all of the re-entrancies (see *18.6. Re-entrant Feature Structures*) that FS does. (Carpenter (1992); see also Pereira (1987) and Shieber (1986)) A more formal definition is provided in ISO 24610-1:2006 .

Following the spirit of the informal definition above, we can extend subsumption in a straightforward way to cover alternation, negation, special primitive values, and the use of attributes in the markup. For instance, a <vAlt> containing the value **v** subsumes **v**. The negation of a value **v** (represented by means of the <vNot> element discussed in section *18.8.2. Negation*) subsumes any value that is not **v**; for example `<vNot><numeric value='0'/></vNot>` subsumes any numeric value other than zero. The value **<fs type="X"/>** subsumes any feature structure of type **X**, even if it is not valid.

As an example of feature declarations, consider the following extract from Gazdar et al.'s *Generalized Phrase Structure Grammar*. In the appendix to their book, they propose a feature system for English of which this is just a sampling:

```
feature value range
INV {+, -}
CONJ {and, both, but, either, neither, nor, or, NIL}
COMP {for, that, whether, if, NIL}
AGR CAT
PFORM {to, by, for, ...}
```

Source: [82]

```
Feature specification defaults
FSD 1: [-INV]
FSD 2: ~[CONJ]
FSD 9: [INF, +SUBJ] --> [COMP for]
```

The INV feature, which encodes whether or not a sentence is inverted, allows only the values plus (+) and minus (-). If the feature is not specified, then the default rule (FSD 1 above) says that a value of minus is always assumed. The feature declaration for this feature would be encoded as follows:

```
<fDecl name="INV">
  <fDescr>inverted sentence</fDescr>
  <vRange>
    <vAlt>
      <binary value="true"/>
      <binary value="false"/>
    </vAlt>
  </vRange>
  <vDefault>
    <binary value="false"/>
  </vDefault>
</fDecl>
```

The value range is specified as an alternation (more precisely, an exclusive disjunction), which can be represented by the <binary> feature value. That is, the value must be either true or false, but cannot be both or neither.

The CONJ feature indicates the surface form of the conjunction used in a construction. The ~ in the default rule (see FSD 2 above) represents negation. This means that by default the feature is not applicable, in other words, no conjunction is taking place. Note that CONJ not being present is distinct from CONJ being present but having the NIL value allowed in the value range. In their analysis, NIL means that the phenomenon of conjunction is taking place but there is no explicit conjunction in the surface form of the sentence. The feature declaration for this feature would be encoded as follows:

```
<fDecl name="CONJ">
  <fDescr>surface form of the conjunction</fDescr>
  <vRange>
    <vAlt>
      <symbol value="and"/>
      <symbol value="both"/>
      <symbol value="but"/>
      <symbol value="either"/>
      <symbol value="neither"/>
      <symbol value="nor"/>
```

```
    <symbol value="or"/>
    <symbol value="NIL"/>
   </vAlt>
  </vRange>
  <vDefault>
   <binary value="false"/>
  </vDefault>
</fDecl>
```

Note that the <vDefault> is not strictly necessary in this case, since the binary value of **false** only serves to convey the information that the feature has no other legitimate value.

The COMP feature indicates the surface form of the complementizer used in a construction. In value range, it is analogous to CONJ. However, its default rule (see FSD 9 above) is conditional. It says that if the verb form is infinitival (the VFORM feature is not mentioned in the rule since it is the only feature that can take INF as a value), and the construction has a subject, then a *for* complement must be used. For instance, to make John the subject of the infinitive in *It is necessary to go,* a *for* complement must be used; that is, *It is necessary for John to go.* The feature declaration for this feature would be encoded as follows:

```
<fDecl name="COMP">
  <fDescr>surface form of the complementizer</fDescr>
  <vRange>
   <vAlt>
    <symbol value="for"/>
    <symbol value="that"/>
    <symbol value="whether"/>
    <symbol value="if"/>
    <symbol value="NIL"/>
   </vAlt>
  </vRange>
  <vDefault>
   <if>
    <fs>
     <f name="VFORM">
      <symbol value="INF"/>
     </f>
     <f name="SUBJ">
      <binary value="true"/>
     </f>
    </fs>
    <then/>
    <symbol value="for"/>
   </if>
  </vDefault>
</fDecl>
```

The AGR feature stores the features relevant to subject-verb agreement. Gazdar et al. specify the range of this feature as CAT. This means that the value is a category, which is their term for a feature structure. This is actually too weak a statement. Not just any feature structure is allowable here; it must be a feature structure for agreement (which is defined in the complete example at the end of the chapter to contain the features of person and number). The following feature declaration encodes this constraint on the value range:

```
<fDecl  name="AGR">
  <fDescr>agreement for person and number</fDescr>
  <vRange>
    <fs  type="Agreement"/>
  </vRange>
</fDecl>
```

That is, the value must be a feature structure of type **Agreement**. The complete example at the end of this chapter includes the **<fsDecl type="Agreement">** which includes **<fDecl name="PERS">** and **<fDecl name="NUM">**.

The PFORM feature indicates the surface form of the preposition used in a construction. Since PFORM is specified above as an open set, <string> is used in the range specification below rather than <symbol>.

```
<fDecl  name="PFORM">
  <fDescr>word form of a preposition</fDescr>
  <vRange>
    <vNot>
      <string/>
    </vNot>
  </vRange>
</fDecl>
```

This example makes use of a negated value: `<vNot><string/></vNot>` subsumes any string that is not the empty string.

Note that the class model.featureVal includes all possible single feature values, including feature structures, alternations (<vAlt>) and complex collections (<vColl>).

18.11.4 Feature Structure Constraints

Ensuring the validity of feature structures may require much more than simply specifying the range of allowed values for each feature. There may be constraints on the co-occurrence of one feature value with the value of another feature in the same feature structure or in an embedded feature structure.

Such constraints on valid feature structures are expressed as a series of conditional and biconditional tests in the <fsConstraints> part of an <fsDecl>. A particular feature structure is valid only if it meets all the constraints. The <cond> element encodes the conventional if-then conditional of boolean logic which succeeds when both the antecedent and consequent are true, or whenever the antecedent is false. The <bicond> element encodes the biconditional (if and only if) operation of boolean logic. It succeeds only when the corresponding if-then conditionals in both directions are true. In feature structure constraints the antecedent and consequent are expressed as feature structures; they are considered true if they subsume (see section *18.11.3. Feature Declarations*) the feature structure in question, but in the case of consequents, this truth is asserted rather than simply tested. That is to say, a conditional is enforced by determining that the antecedent does not (and will never) subsume the given feature structure, or by determining that the antecedent does subsume the given feature structure, and then unifying the consequent with it (the result of which, if successful, will be subsumed by the consequent). In practice, the enforcement of such constraints can result in periods in which the truth of a constraint with respect to a given feature structure is simply not known; in this case, the constraint must be persistently monitored as the feature structure becomes more informative until either its truth value is determined or computation fails for some other reason.

The following elements make up the <fsConstraints> part of an FSD:

<fsConstraints> (feature-structure constraints) specifies constraints on the content of valid feature structures.

<cond> (conditional feature-structure constraint) defines a conditional feature-structure constraint; the consequent and the antecedent are specified as feature structures or feature-structure collections; the constraint is satisfied if both the antecedent and the consequent subsume a given feature structure, or if the antecedent does not.

<bicond> (bi-conditional feature-structure constraint) defines a biconditional feature-structure constraint; both consequent and antecedent are specified as feature structures or groups of feature structures; the constraint is satisfied if both subsume a given feature structure, or if both do not.

<then/> separates the condition from the default in an <if>, or the antecedent and the consequent in a <cond> element.

<iff/> (if and only if) separates the condition from the consequence in a bicond element.

For an example of feature structure constraints, consider the following 'feature co-occurrence restrictions' extracted from the feature system for English proposed by Gazdar, et al. (1985:246–247):

```
    [FCR 1:   [+INV] → [+AUX, FIN]
```

```
    FCR 7:   [BAR 0] ≡ [N] & [V] & [SUBCAT]
```

```
    FCR 8:   [BAR 1] → ~[SUBCAT]]
```

The first constraint says that if a construction is inverted, it must also have an auxiliary and a finite verb form. That is,

```
<cond>
  <fs>
    <f  name="INV">
      <binary  value="true"/>
    </f>
  </fs>
  <then/>
  <fs>
    <f  name="AUX">
      <binary  value="true"/>
    </f>
    <f  name="VFORM">
      <symbol  value="FIN"/>
    </f>
  </fs>
</cond>
```

The second constraint says that if a construction has a BAR value of zero (i.e., it is a sentence), then it must have a value for the features N, V, and SUBCAT. By the same token, because it is a biconditional, if it has values for N, V, and SUBCAT, it must have BAR='0'. That is,

```
<bicond>
  <fs>
    <f  name="BAR">
```

```
    <symbol value="0"/>
   </f>
  </fs>
  <iff/>
  <fs>
   <f name="N">
    <binary value="true"/>
   </f>
   <f name="V">
    <binary value="true"/>
   </f>
   <f name="SUBCAT">
    <binary value="true"/>
   </f>
  </fs>
</bicond>
```

The final constraint says that if a construction has a BAR value of 1 (i.e., it is a phrase), then the SUBCAT feature should be absent (~). This is not biconditional, since there are other instances under which the SUBCAT feature is inappropriate. That is,

```
<cond>
  <fs>
   <f name="BAR">
    <symbol value="1"/>
   </f>
  </fs>
  <then/>
  <fs>
   <f name="SUBCAT">
    <binary value="false"/>
   </f>
  </fs>
</cond>
```

Note that <cond> and <bicond> use the empty tags <then> and <iff>, respectively, to separate the antecedent and consequent. These are primarily for the sake of enhancing human readability.

18.11.5 A Complete Example

To summarize this chapter, the complete FSD for the example that has run through the chapter is reproduced below:

```
<TEI>
  <teiHeader>
   <fileDesc>
    <titleStmt>
     <title>A sample FSD based on an extract from Gazdar
            et al.'s GPSG feature system for English</title>
     <respStmt>
      <resp>encoded by</resp>
      <name>Gary F. Simons</name>
     </respStmt>
    </titleStmt>
```

```
    <publicationStmt>
      <p>This sample was first encoded by Gary F. Simons (Summer
             Institute of Linguistics, Dallas, TX) on January 28, 1991.
             Revised April 8, 1993 to match the specification of FSDs
             in version P2 of the TEI Guidelines. Revised again December 2004 to
             be consistent with the feature structure representation standard
             jointly developed with ISO TC37/SC4.
      </p>
    </publicationStmt>
    <sourceDesc>
      <p>This sample FSD does not describe a complete feature
             system. It is based on extracts from the feature system
             for English presented in the appendix (pages 245–247) of
             Generalized Phrase Structure Grammar, by Gazdar, Klein,
             Pullum, and Sag (Harvard University Press, 1985).</p>
    </sourceDesc>
  </fileDesc>
</teiHeader>
<fsdDecl>
  <fsDecl type="GPSG">
    <fsDescr>Encodes a feature structure for the GPSG analysis
          of English (after Gazdar, Klein, Pullum, and Sag)</fsDescr>
    <fDecl name="INV">
      <fDescr>inverted sentence</fDescr>
      <vRange>
        <vAlt>
          <binary value="true"/>
          <binary value="false"/>
        </vAlt>
      </vRange>
      <vDefault>
        <binary value="false"/>
      </vDefault>
    </fDecl>
    <fDecl name="CONJ">
      <fDescr>surface form of the conjunction</fDescr>
      <vRange>
        <vAlt>
          <symbol value="and"/>
          <symbol value="both"/>
          <symbol value="but"/>
          <symbol value="either"/>
          <symbol value="neither"/>
          <symbol value="nor"/>
          <symbol value="or"/>
          <symbol value="NIL"/>
        </vAlt>
      </vRange>
      <vDefault>
        <binary value="false"/>
      </vDefault>
    </fDecl>
    <fDecl name="COMP">
      <fDescr>surface form of the complementizer</fDescr>
      <vRange>
        <vAlt>
          <symbol value="for"/>
```

```
      <symbol value="that"/>
      <symbol value="whether"/>
      <symbol value="if"/>
      <symbol value="NIL"/>
    </vAlt>
  </vRange>
  <vDefault>
    <if>
      <fs>
        <f name="VFORM">
          <symbol value="INF"/>
        </f>
        <f name="SUBJ">
          <binary value="true"/>
        </f>
      </fs>
      <then/>
      <symbol value="for"/>
    </if>
  </vDefault>
</fDecl>
<fDecl name="AGR">
  <fDescr>agreement for person and number</fDescr>
  <vRange>
    <fs type="Agreement"/>
  </vRange>
</fDecl>
<fDecl name="PFORM">
  <fDescr>word form of a preposition</fDescr>
  <vRange>
    <vNot>
      <string/>
    </vNot>
  </vRange>
</fDecl>
<fsConstraints>
  <cond>
    <fs>
      <f name="INV">
        <binary value="true"/>
      </f>
    </fs>
    <then/>
    <fs>
      <f name="AUX">
        <binary value="true"/>
      </f>
      <f name="VFORM">
        <symbol value="FIN"/>
      </f>
    </fs>
  </cond>
  <bicond>
    <fs>
      <f name="BAR">
        <symbol value="0"/>
      </f>
```

```
        </fs>
        <iff/>
        <fs>
          <f name="N">
            <binary value="true"/>
          </f>
          <f name="V">
            <binary value="true"/>
          </f>
          <f name="SUBCAT">
            <binary value="true"/>
          </f>
        </fs>
      </bicond>
      <cond>
        <fs>
          <f name="BAR">
            <symbol value="1"/>
          </f>
        </fs>
        <then/>
        <fs>
          <f name="SUBCAT">
            <binary value="false"/>
          </f>
        </fs>
      </cond>
    </fsConstraints>
  </fsDecl>
  <fsDecl type="Agreement">
    <fsDescr>This type of feature structure encodes the features
          for subject-verb agreement in English</fsDescr>
    <fDecl name="PERS">
      <fDescr>person (first, second, or third)</fDescr>
      <vRange>
        <vAlt>
          <symbol value="1"/>
          <symbol value="2"/>
          <symbol value="3"/>
        </vAlt>
      </vRange>
    </fDecl>
    <fDecl name="NUM">
      <fDescr>number (singular or plural)</fDescr>
      <vRange>
        <vAlt>
          <symbol value="sg"/>
          <symbol value="pl"/>
        </vAlt>
      </vRange>
    </fDecl>
  </fsDecl>
 </fsdDecl>
</TEI>
```

18.12 Formal Definition and Implementation

This elements discussed in this chapter constitute a module of the TEI scheme which is formally defined as follows:

Module iso-fs: Feature structures

- Elements defined: bicond binary cond default f fDecl fDescr fLib fs fsConstraints fsDecl fsDescr fsdDecl fsdLink fvLib if iff numeric string symbol then vAlt vColl vDefault vLabel vMerge vNot vRange

The selection and combination of modules to form a TEI schema is described in *1.2. Defining a TEI Schema*.

Chapter 19

Graphs, Networks, and Trees

Graphical representations are widely used for displaying relations among informational units because they help readers to visualize those relations and hence to understand them better. Two general types of graphical representations may be distinguished.

- Graphs, in the strictly mathematical sense, consist of points, often called nodes or vertices, and connections among them, called arcs, or under certain conditions, edges. Among the various types of graphs are networks and trees. Graphs generally and networks in particular are dealt with directly below. Trees are dealt with separately in sections *19.2. Trees* and *19.3. Another Tree Notation*.[1]
- Charts, which typically plot data in two or more dimensions, including plots with orthogonal or radial axes, bar charts, pie charts, and the like. These can be described using the elements defined in the module for figures and graphics; see chapter *14. Tables, Formulæ, and Graphics*.

Among the types of qualitative relations often represented by graphs are organizational hierarchies, flow charts, genealogies, semantic networks, transition networks, grammatical relations, tournament schedules, seating plans, and directions to people's houses. In developing recommendations for the encoding of graphs of various types, we have relied on their formal mathematical definitions and on the most common conventions for representing them visually. However, it must be emphasized that these recommendations do not provide for the full range of possible graphical representations, and deal only partially with questions of design, layout, and placement.

19.1 Graphs and Digraphs

Broadly speaking, graphs can be divided into two types: undirected and directed. An undirected graph is a set of nodes (or vertices) together with a set of pairs of those vertices, called arcs or edges. Each node in an arc of an undirected graph is said to be incident with that arc, and the two vertices (nodes) which make up an arc are said to be adjacent. An directed graph is like an undirected graph except that the arcs are ordered pairs of nodes. In the case of directed graphs, the term edge is not used; moreover, each arc in a directed graph is said to be adjacent from the node from which the arc emanates, and adjacent to the node to which the arc is directed. We use the element <graph> to encode graphs as a whole, <node> to encode nodes or vertices, and <arc> to encode arcs or edges; arcs can also be encoded by attributes on the <node> element. These elements have the following descriptions and attributes:

<graph> encodes a graph, which is a collection of nodes, and arcs which connect the nodes.

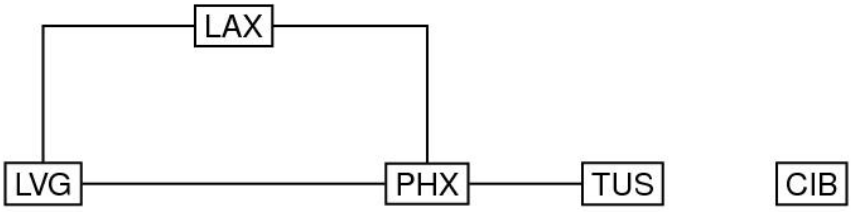
<node> encodes a node, a possibly labeled point in a graph.

<arc> encodes an arc, the connection from one node to another in a graph.

[1] The treatment here is largely based on the characterizations of graph types in Chartrand and Lesniak (1986)

Before proceeding, some additional terminology may be helpful. We define a path in a graph as a sequence of nodes n1, ..., nk such that there is an arc from each ni to ni+1 in the sequence. A cyclic path, or cycle is a path leading from a particular node back to itself. A graph that contains at least one cycle is said to be cyclic; otherwise it is acyclic. We say, finally, that a graph is connected if there is a path from some node to every other node in the graph; any graph that is not connected is said to be disconnected.

Here is an example of an undirected, cyclic disconnected graph, in which the nodes are annotated with three-letter codes for airports, and the arcs connecting the nodes are represented by horizontal and vertical lines, with 90 degree bends used simply to avoid having to draw diagonal lines.



Next is a markup of the graph, using <arc> elements to encode the arcs.

```
<graph
   type="undirected"
   xml:id="CUG1"
   order="5"
   size="4">
  <label>Airline Connections in Southwestern USA</label>
  <node  xml:id="LAX"  degree="2">
    <label>LAX</label>
  </node>
  <node  xml:id="LVG"  degree="2">
    <label>LVG</label>
  </node>
  <node  xml:id="PHX"  degree="3">
    <label>PHX</label>
  </node>
  <node  xml:id="TUS"  degree="1">
    <label>TUS</label>
  </node>
  <node  xml:id="CIB"  degree="0">
    <label>CIB</label>
  </node>
  <arc  from="#LAX"  to="#LVG"/>
  <arc  from="#LAX"  to="#PHX"/>
  <arc  from="#LVG"  to="#PHX"/>
  <arc  from="#PHX"  to="#TUS"/>
</graph>
```

The first child element of <graph> may be a <label> to record a label for the graph; similarly, the <label> child of each <node> element records the labels of that node. The **order** and **size** attributes on the <graph> element record the number of nodes and number of arcs in the graph respectively; these values are optional (since they can be computed from the rest of the graph), but if they are supplied, they must be consistent with the rest of the encoding. They can thus be used to help check that the graph has been encoded and transmitted correctly. The **degree** attribute on the <node> elements record the number of arcs that are incident with that node. It is optional (because redundant), but can be used to help in validity checking: if a value is given, it must be consistent with the rest of the information in the graph. Finally, the **from** and **to** attributes on the <arc> elements provide pointers to the nodes connected by those arcs. Since the graph is undirected, no directionality

is implied by the use of the **from** and **to** attributes; the values of these attributes could be interchanged in each arc without changing the graph.

The **adj**, **adjFrom**, and **adjTo** attributes of the <node> element provide an alternative method of representing unlabeled arcs, their values being pointers to the nodes which are adjacent to or from that node. The **adj** attribute is to be used for undirected graphs, and the **adjFrom** and **adjTo** attributes for directed graphs. It is a semantic error for the directed adjacency attributes to be used in an undirected graph, and vice versa. Here is a markup of the preceding graph, using the **adj** attribute to represent the arcs.

```
<graph
   type="undirected"
   xml:id="CUG2"
   order="5"
   size="4">
  <label>Airline Connections in Southwestern USA</label>
  <node  xml:id="LAX2"  degree="2"  adj="#LVG2 #PHX2">
    <label>LAX2</label>
  </node>
  <node  xml:id="LVG2"  degree="2"  adj="#LAX2 #PHX2">
    <label>LVG2</label>
  </node>
  <node  xml:id="PHX2"  degree="3"  adj="#LAX2 #LVG2 #TUS2">
    <label>PHX2</label>
  </node>
  <node  xml:id="TUS2"  degree="1"  adj="#PHX2">
    <label>TUS2</label>
  </node>
  <node  xml:id="CIB2"  degree="0">
    <label>CIB2</label>
  </node>
</graph>
```

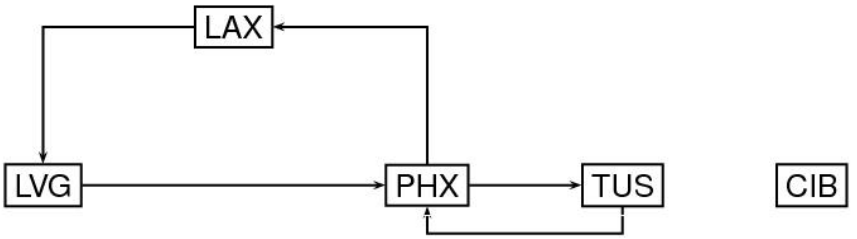
Note that each arc is represented twice in this encoding of the graph. For example, the existence of the arc from LAX to LVG can be inferred from each of the first two <node> elements in the graph. This redundancy, however, is not required: it suffices to describe an arc in any one of the three places it can be described (either adjacent node, or in a separate <arc> element). Here is a less redundant representation of the same graph.

```
<graph
   type="undirected"
   xml:id="CUG3"
   order="5"
   size="4">
  <label>Airline Connections in Southwestern USA</label>
  <node  xml:id="LAX3"  degree="2"  adj="#LVG3 #PHX3">
    <label>LAX3</label>
  </node>
  <node  xml:id="LVG3"  degree="2"  adj="#PHX3">
    <label>LVG3</label>
  </node>
  <node  xml:id="PHX3"  degree="3"  adj="#TUS3">
    <label>PHX3</label>
  </node>
  <node  xml:id="TUS3"  degree="1">
    <label>TUS3</label>
  </node>
```

```
  <node  xml:id="CIB3"  degree="0">
    <label>CIB3</label>
  </node>
</graph>
```

Although in many cases the <arc> element is redundant (since arcs can be described using the adjacency attributes of their adjacent nodes), it has nevertheless been included in this module, in order to allow the convenient specification of identifiers, display or rendition information, and labels for each arc (using the attributes **xml:id**, **rend**, and a child <label> element).

Next, let us modify the preceding graph by adding directionality to the arcs. Specifically, we now think of the arcs as specifying selected routes from one airport to another, as indicated by the direction of the arrowheads in the following diagram.



Here is an encoding of this graph, using the <arc> element to designate the arcs.

```
<graph
   type="directed"
   xml:id="RDG1"
   order="5"
   size="5">
  <label>Selected Airline Routes in Southwestern USA</label>
  <node  xml:id="LAX4"  inDegree="1"  outDegree="1">
    <label>LAX4</label>
  </node>
  <node  xml:id="LVG4"  inDegree="1"  outDegree="1">
    <label>LVG4</label>
  </node>
  <node  xml:id="PHX4"  inDegree="2"  outDegree="2">
    <label>PHX4</label>
  </node>
  <node  xml:id="TUS4"  inDegree="1"  outDegree="1">
    <label>TUS4</label>
  </node>
  <node  xml:id="CIB4"  inDegree="0"  outDegree="0">
    <label>CIB4</label>
  </node>
  <arc  from="#LAX4"  to="#LVG4"/>
  <arc  from="#LVG4"  to="#PHX4"/>
  <arc  from="#PHX4"  to="#LAX4"/>
  <arc  from="#PHX4"  to="#TUS4"/>
  <arc  from="#TUS4"  to="#PHX4"/>
</graph>
```

The attributes **inDegree** and **outDegree** indicate the number of nodes which are adjacent to and from the node concerned respectively.

Here is another encoding of the graph, using the **adjTo** and **adjFrom** attributes on nodes to designate the arcs.

```
<graph
   type="directed"
   xml:id="RDG2"
   order="5"
   size="5">
  <label>Selected Airline Routes in Southwestern USA</label>
  <node
     xml:id="LAX5"
     inDegree="1"
     outDegree="1"
     adjTo="#LVG5"
     adjFrom="#PHX5">
    <label>LAX5</label>
  </node>
  <node
     xml:id="LVG5"
     inDegree="1"
     outDegree="1"
     adjFrom="#LAX5"
     adjTo="#PHX5">
    <label>LVG5</label>
  </node>
  <node
     xml:id="PHX5"
     inDegree="2"
     outDegree="2"
     adjTo="#LAX5 #TUS"
     adjFrom="#LVG5 #TUS5">
    <label>PHX5</label>
  </node>
  <node
     xml:id="TUS5"
     inDegree="1"
     outDegree="1"
     adjTo="#PHX5"
     adjFrom="#PHX5">
    <label>TUS5</label>
  </node>
  <node  xml:id="CIB5"  inDegree="0"  outDegree="0">
    <label>CIB5</label>
  </node>
</graph>
```

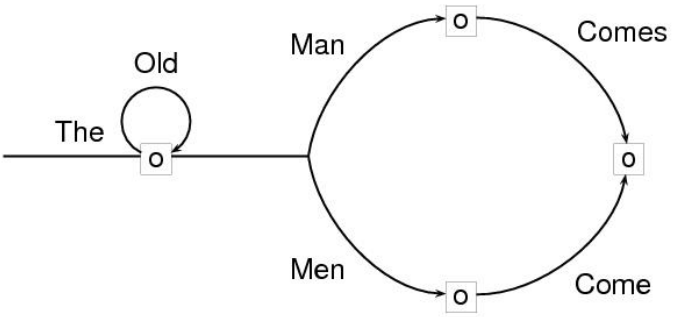
If we wish to label the arcs, say with flight numbers, then <arc> elements must be used to hold the <label> elements, as in the following example.

```
<graph
   type="directed"
   xml:id="RDG3"
   order="5"
   size="5">
  <label>Selected Airline Routes in Southwestern USA</label>
  <node  xml:id="LAX6">
```

```
    <label>LAX6</label>
  </node>
  <node  xml:id="LVG6">
    <label>LVG6</label>
  </node>
  <node  xml:id="PHX6">
    <label>PHX6</label>
  </node>
  <node  xml:id="TUS6">
    <label>TUS6</label>
  </node>
  <node  xml:id="CIB6">
    <label>CIB6</label>
  </node>
  <arc  from="#LAX6"  to="#LVG6">
    <label>SW117</label>
  </arc>
  <arc  from="#LVG6"  to="#PHX6">
    <label>SW711</label>
  </arc>
  <arc  from="#PHX6"  to="#LAX6">
    <label>AA218</label>
  </arc>
  <arc  from="#PHX6"  to="#TUS6">
    <label>AW229</label>
  </arc>
  <arc  from="#TUS6"  to="#PHX6">
    <label>AW225</label>
  </arc>
</graph>
```

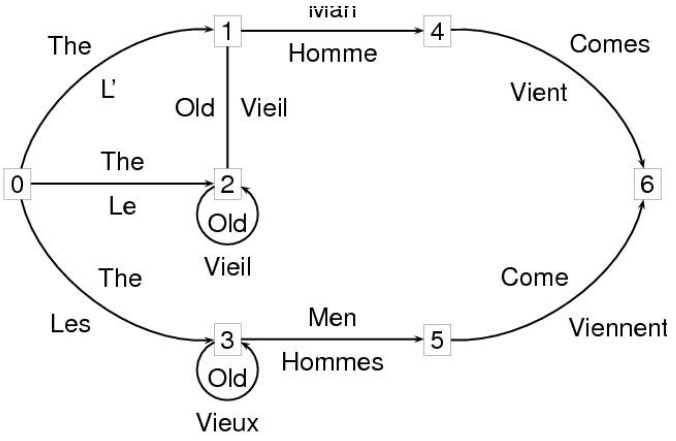
19.1.1 Transition Networks

For encoding transition networks and other kinds of directed graphs in which distinctions among types of nodes must be made, the **type** attribute is provided for <node> elements. In the following example, the initial and final nodes (or states) of the network are distinguished. It can be understood as accepting the set of strings obtained by traversing it from its initial node to its final node, and concatenating the labels.



```
<graph
   type="network-transition"
   xml:id="SS8"
   order="5"
   size="6">
  <label>(8)</label>
  <node
     xml:id="Q0"
     inDegree="0"
     outDegree="1"
     type="initial"/>
  <node  xml:id="Q1"  inDegree="2"  outDegree="3"/>
  <node  xml:id="Q2"  inDegree="1"  outDegree="1"/>
  <node  xml:id="Q3"  inDegree="1"  outDegree="1"/>
  <node
     xml:id="Q4"
     inDegree="2"
     outDegree="0"
     type="final"/>
  <arc  from="#Q0"  to="#Q1">
    <label>THE</label>
  </arc>
  <arc  from="#Q1"  to="#Q1">
    <label>OLD</label>
  </arc>
  <arc  from="#Q1"  to="#Q2">
    <label>MAN</label>
  </arc>
  <arc  from="#Q1"  to="#Q3">
    <label>MEN</label>
  </arc>
  <arc  from="#Q2"  to="#Q4">
    <label>COMES</label>
  </arc>
  <arc  from="#Q3"  to="#Q4">
    <label>COME</label>
  </arc>
</graph>
```

A finite state transducer has two labels on each arc, and can be thought of as representing a mapping from one sequence of labels to the other. The following example represents a transducer for translating the English strings accepted by the network in the preceding example into French. The nodes have been annotated with numbers, for convenience.

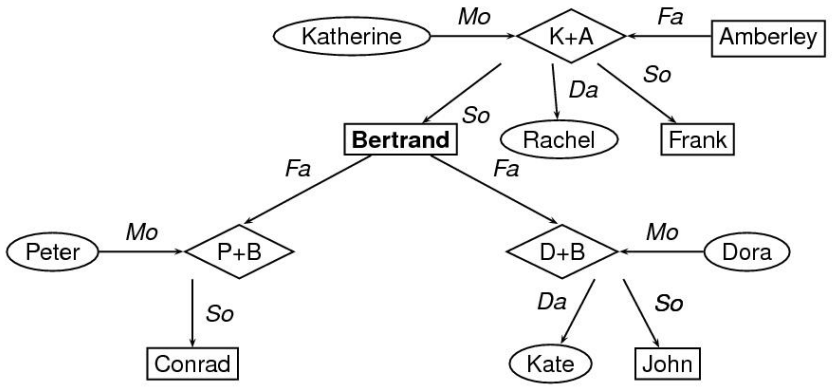



```
<graph type="transducer" order="7" size="10">
 <node
    xml:id="T0"
    inDegree="0"
    outDegree="3"
    type="initial">
  <label>0</label>
 </node>
 <node xml:id="T1" inDegree="2" outDegree="1">
  <label>1</label>
 </node>
 <node xml:id="T2" inDegree="2" outDegree="2">
  <label>2</label>
 </node>
 <node xml:id="T3" inDegree="2" outDegree="2">
  <label>3</label>
 </node>
 <node xml:id="T4" inDegree="1" outDegree="1">
  <label>4</label>
 </node>
 <node xml:id="T5" inDegree="1" outDegree="1">
  <label>5</label>
 </node>
 <node
    xml:id="T6"
    inDegree="2"
    outDegree="0"
    type="final">
  <label>6</label>
 </node>
 <arc from="#T0" to="#T1">
  <label>THE</label>
  <label>L'</label>
 </arc>
 <arc from="#T0" to="#T2">
  <label>THE</label>
  <label>LE</label>
 </arc>
 <arc from="#T0" to="#T3">
```

```
    <label>THE</label>
    <label>LES</label>
  </arc>
  <arc  from="#T1"  to="#T4">
    <label>MAN</label>
    <label>HOMME</label>
  </arc>
  <arc  from="#T2"  to="#T1">
    <label>OLD</label>
    <label>VIEIL</label>
  </arc>
  <arc  from="#T2"  to="#T2">
    <label>OLD</label>
    <label>VIEIL</label>
  </arc>
  <arc  from="#T3"  to="#T3">
    <label>OLD</label>
    <label>VIEUX</label>
  </arc>
  <arc  from="#T3"  to="#T5">
    <label>MEN</label>
    <label>HOMMES</label>
  </arc>
  <arc  from="#T4"  to="#T6">
    <label>COMES</label>
    <label>VIENT</label>
  </arc>
  <arc  from="#T5"  to="#T6">
    <label>COME</label>
    <label>VIENNENT</label>
  </arc>
</graph>
```

19.1.2 Family Trees

The next example provides an encoding a portion of a family tree, in which nodes are used to represent individuals and parents of individuals, and arcs are used to represent common parentage and descent links. Let us suppose, further, that information about individuals is contained in feature structures, which are contained in feature-structure libraries elsewhere (see *18.4. Feature and Feature-Value Libraries*). We can use the **value** attribute on <node> elements to point to those feature structures. In this particular representation of the graph, nodes representing females are framed by ovals, nodes representing males are framed by boxes, and nodes representing parents are framed by diamonds.




```
<graph type="family_tree" order="13" size="12">
  <node
     xml:id="KATHR"
     value="http://example.com/russell-fs/tei/kr1"
     inDegree="0"
     outDegree="1">
    <label>Katherine</label>
  </node>
  <node
     xml:id="AMBER"
     value="http://example.com/russell-fs/tei/ar1"
     inDegree="0"
     outDegree="1">
    <label>Amberley</label>
  </node>
  <node xml:id="KAR" inDegree="2" outDegree="3">
    <label>K+A</label>
  </node>
  <node
     xml:id="BERTR"
     value="http://example.com/russell-fs/tei/br1"
     inDegree="1"
     outDegree="2">
    <label>Bertrand</label>
  </node>
  <node
     xml:id="PETER"
     value="http://example.com/russell-fs/tei/pr1"
     inDegree="0"
     outDegree="1">
    <label>Peter</label>
  </node>
  <node
     xml:id="DORAR"
     value="http://example.com/russell-fs/tei/dr1"
     inDegree="0"
     outDegree="1">
    <label>Dora</label>
  </node>
  <node xml:id="PBR" inDegree="2" outDegree="1">
    <label>P+B</label>
  </node>
```

```
<node xml:id="DBR" inDegree="2" outDegree="2">
  <label>D+B</label>
</node>
<node
    xml:id="FRANR"
    value="http://example.com/russell-fs/tei/fr1"
    inDegree="1"
    outDegree="0">
  <label>Frank</label>
</node>
<node
    xml:id="RACHR"
    value="http://example.com/russell-fs/tei/rr1"
    inDegree="1"
    outDegree="0">
  <label>Rachel</label>
</node>
<node
    xml:id="CONRR"
    value="http://example.com/russell-fs/tei/cr1"
    inDegree="1"
    outDegree="0">
  <label>Conrad</label>
</node>
<node
    xml:id="KATER"
    value="http://example.com/russell-fs/tei/kr2"
    inDegree="1"
    outDegree="0">
  <label>Kate</label>
</node>
<node
    xml:id="JOHNR"
    value="http://example.com/russell-fs/tei/jr1"
    inDegree="1"
    outDegree="0">
  <label>John</label>
</node>
<arc from="#KATHR" to="#KAR">
  <label>Mo</label>
</arc>
<arc from="#AMBER" to="#KAR">
  <label>Fa</label>
</arc>
<arc from="#KAR" to="#BERTR">
  <label>So</label>
</arc>
<arc from="#KAR" to="#FRANR">
  <label>So</label>
</arc>
<arc from="#KAR" to="#RACHR">
  <label>Da</label>
</arc>
<arc from="#PETER" to="#PBR">
  <label>Mo</label>
</arc>
<arc from="#BERTR" to="#PBR">
```

```
    <label>Fa</label>
  </arc>
  <arc  from="#PBR"  to="#CONRR">
    <label>So</label>
  </arc>
  <arc  from="#DORAR"  to="#DBR">
    <label>Mo</label>
  </arc>
  <arc  from="#BERTR"  to="#DBR">
    <label>Fa</label>
  </arc>
  <arc  from="#DBR"  to="#KATER">
    <label>Da</label>
  </arc>
  <arc  from="#DBR"  to="#JOHNR">
    <label>So</label>
  </arc>
</graph>
```

Source: [154]

19.1.3 Historical Interpretation

For our final example, we represent graphically the relationships among various geographic areas mentioned in a seventeenth-century Scottish document. The document itself is a 'sasine', which records a grant of land from the earl of Argyll to one Donald McNeill, and reads in part as follows (abbreviations have been expanded silently, and '[...]' marks illegible passages):

> Item instrument of Sasine given the said Hector Mcneil confirmed and dated 28 May 1632 [...] at Edinburgh upon the 15 June 1632
>
> Item ane charter granted by Archibald late earl of Argyle and Donald McNeill of Gallachalzie wh makes mention that ... the said late Earl yields and grants to the said Donald MacNeill ...
>
> All and hail the two merk land of old extent of Gallachalzie with the pertinents by and in the lordship of Knapdale within the sherrifdome of Argyll
>
> [description of other lands granted follows ...]
>
> This Charter is dated at Inverary the 15th May 1669

In this example, we are concerned with the land and pertinents (i.e. accompanying sources of revenue) described as 'the two merk land of old extent of Gallachalzie with the pertinents by and in the lordship of Knapdale within the sherrifdom of Argyll'.

The passage concerns the following pieces of land:

- the Earl of Argyll's land (i.e. the lands granted by this clause of the sasine)
- two mark of land in Gallachalzie
- the pertinents for this land
- the Lordship of Knapdale
- the sherrifdom of Argyll

We will represent these geographic entities as nodes in a graph. Arcs in the graph will represent the following relationships among them:

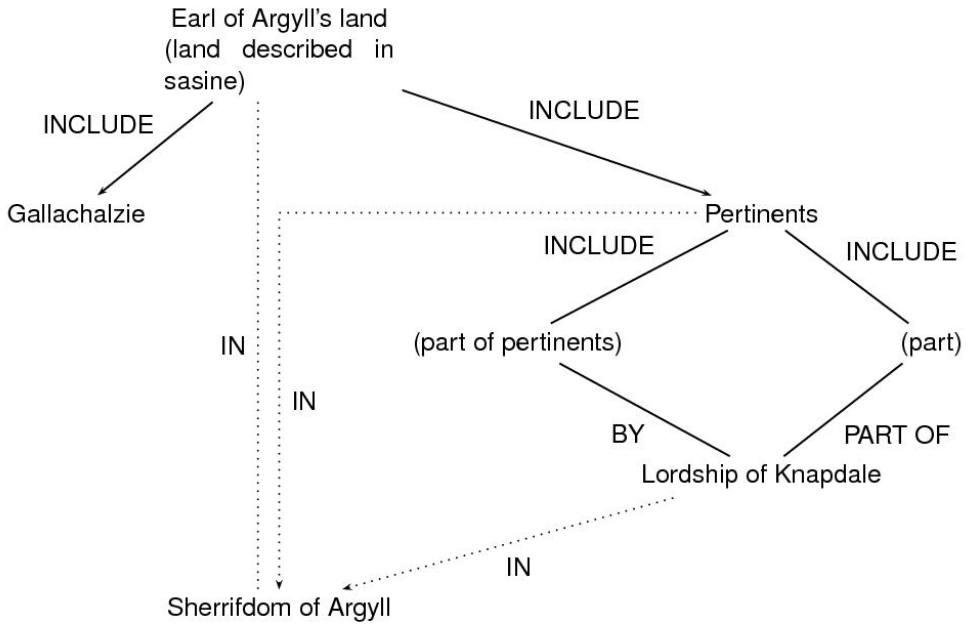
- containment (INCLUDE)

- location within (IN)
- contiguity (BY)
- constituency (PART OF)

Note that these relationships are logically related: 'include' and 'in', for example, are inverses of each other: the Earl of Argyll's land includes the parcel in Gallachalzie, and the parcel is therefore in the Earl of Argyll's land. Given an explicit set of inference rules, an appropriate application could use the graph we are constructing to infer the logical consequences of the relationships we identify.

Let us assume that feature-structure analyses are available which describe Gallachalzie, Knapdale, and Argyll. We will link to those feature structures using the **value** attribute on the nodes representing those places. However, there may be some uncertainty as to which noun phrase is modified by the phrase 'within the sheriffdome of Argyll': perhaps the entire lands (land and pertinents) are in Argyll, perhaps just the pertinents are, or perhaps only Knapdale is (together with the portion of the pertinents which is in Knapdale). We will represent all three of these interpretations in the graph; they are, however, mutually exclusive, which we represent using the **exclude** attribute defined in chapter *16. Linking, Segmentation, and Alignment*.[2]

We represent the graph and its encoding as follows, where the dotted lines in the graph indicate the mutually exclusive arcs; in the encoding, we use the **exclude** attribute to indicate those arcs.



The graph formalizes the following relationships:

- the Earl of Argyll's land *includes* (the parcel of land in) Gallachalzie
- the Earl of Argyll's land *includes* the pertinents of that parcel
- the pertinents are (in part) *by* the Lordship of Knapdale
- the pertinents are (in part) *part of* the Lordship of Knapdale

[2] That is, the three syntactic interpretations of the clause are mutually exclusive. The notion that the pertinents are in Argyll is clearly not inconsistent with the notion that both the land in Gallachalzie and the pertinents are in Argyll. The graph given here describes the possible interpretations of the clause itself, not the sets of inferences derivable from each syntactic interpretation, for which it would be convenient to use the facilities described in chapter *18. Feature Structures*.

- the Earl of Argyll's land, or the pertinents, or the Lordship of Knapdale, is *in* the Sherrifdom of Argyll
  We encode the graph thus:

```
<graph type="directed" order="7" size="9">
 <node xml:id="EARL">
  <label>Earl of Argyll's land</label>
 </node>
 <node xml:id="GALL"
    value="http://example.com/people/scots#gall">
  <label>Gallachalzie</label>
 </node>
 <node xml:id="PERT">
  <label>Pertinents</label>
 </node>
 <node xml:id="PER1">
  <label>Pertinents part</label>
 </node>
 <node xml:id="PER2">
  <label>Pertinents part</label>
 </node>
 <node
    xml:id="KNAP"
    value="http://example.com/people/scots#knapfs">
  <label>Lordship of Knapdale</label>
 </node>
 <node
    xml:id="ARGY"
    value="http://example.com/people/scots#argyfs">
  <label>Sherrifdome of Argyll</label>
 </node>
 <arc xml:id="EARLGALL" from="#EARL" to="#GALL">
  <label>INCLUDE</label>
 </arc>
 <arc
    xml:id="EARLARGY"
    from="#EARL"
    to="#ARGY"
    exclude="#PERTARGY #KNAPARGY">
  <label>IN</label>
 </arc>
 <arc xml:id="EARLPERT" from="#EARL" to="#PERT">
  <label>INCLUDE</label>
 </arc>
 <arc xml:id="PERTPER1" from="#PERT" to="#PER1">
  <label>INCLUDE</label>
 </arc>
 <arc xml:id="PERTPER2" from="#PERT" to="#PER2">
  <label>INCLUDE</label>
 </arc>
 <arc
    xml:id="PERTARGY"
    from="#PERT"
    to="#ARGY"
    exclude="#EARLARGY #KNAPARGY">
  <label>IN</label>
 </arc>
 <arc xml:id="PER1KNAP" from="#PER1" to="#KNAP">
```

```
    <label>BY</label>
  </arc>
  <arc  xml:id="PER2KNAP"  from="#PER2"  to="#KNAP">
    <label>PART OF</label>
  </arc>
  <arc
     xml:id="KNAPARGY"
     from="#KNAP"
     to="#ARGY"
     exclude="#EARLARGY #PERTARGY">
    <label>IN</label>
  </arc>
</graph>
```

19.2 Trees

A tree is a connected acyclic graph. That is, it is possible in a tree graph to follow a path from any vertex to any other vertex, but there are no paths that lead from any vertex to itself. A rooted tree is a directed graph based on a tree; that is, the arcs in the graph correspond to the arcs of a tree such that there is exactly one node, called the root, for which there is a path from that node to all other nodes in the graph. For our purposes, we may ignore all trees except for rooted trees, and hence we shall use the <tree> element for rooted trees, and the <root> element for its root. The nodes adjacent to a given node are called its children, and the node adjacent from a given node is called its parent. Nodes with both a parent and children are called internal nodes, for which we use the <iNode> element. A node with no children is tagged as a <leaf>. If the children of a node are ordered from left to right, then we say that that node is ordered. If all the nodes of a tree are ordered, then we say that the tree is an ordered tree. If some of the nodes of a tree are ordered and others are not, then the tree is a partially ordered tree. The ordering of nodes and trees may be specified by an attribute; we take the default ordering for trees to be ordered, that roots inherit their ordering from the trees in which they occur, and internal nodes inherit their ordering from their parents. Finally, we permit a node to be specified as following other nodes, which (when its parent is ordered) it would be assumed to precede, giving rise to crossing arcs. The elements used for the encoding of trees have the following descriptions and attributes.

<tree> encodes a tree, which is made up of a root, internal nodes, leaves, and arcs from root to leaves.

- @**arity** gives the maximum number of children of the root and internal nodes of the tree.
- @**ord** (ordered) indicates whether or not the tree is ordered, or if it is partially ordered.
- @**order** gives the order of the tree, i.e., the number of its nodes.

<root> (root node) represents the root node of a tree.

- @**value** provides the value of the root, which is a feature structure or other analytic element.
- @**children** provides a list of identifiers of the elements which are the children of the root node.
- @**ord** (ordered) indicates whether or not the root is ordered.
- @**outDegree** gives the out degree of the root, the number of its children.

<iNode> (intermediate (or internal) node) represents an intermediate (or internal) node of a tree.

- @**value** provides the value of an intermediate node, which is a feature structure or other analytic element.
- @**children** provides a list of identifiers of the elements which are the children of the intermediate node.
- @**parent** provides the identifier of the element which is the parent of this node.

@**ord** (ordered) indicates whether or not the internal node is ordered.

@**follow** provides an identifier of the element which this node follows.

@**outDegree** gives the out degree of an intermediate node, the number of its children.

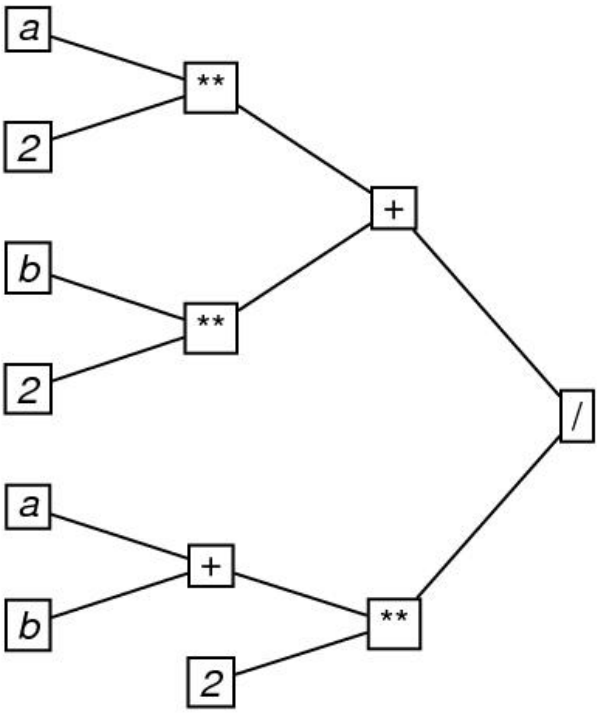
<leaf> encodes the leaves (terminal nodes) of a tree.

@**value** provides a pointer to a feature structure or other analytic element.

@**parent** provides the identifier of parent of a leaf.

@**follow** provides an identifier of an element which this leaf follows.

Here is an example of a tree. It represents the order in which the operators of addition (symbolized by +), exponentiation (symbolized by **) and division (symbolized by /) are applied in evaluating the arithmetic formula `((a**2)+(b**2))/((a+b)**2)`. In drawing the graph, the root is placed on the far right, and directionality is presumed to be to the left.




```
<tree
   n="ex1"
   arity="2"
   ord="true"
   order="12">
  <root  xml:id="G-DIV1"  children="#PLU1 #EXP1">
    <label>/</label>
  </root>
  <iNode  xml:id="PLU1"  parent="#G-DIV1"  children="#EXP2 #EXP3">
    <label>+</label>
  </iNode>
  <iNode  xml:id="EXP1"  parent="#G-DIV1"  children="#PLU2 #NUM2.3">
    <label>**</label>
  </iNode>
  <iNode  xml:id="EXP2"  parent="#PLU1"  children="#VARA1 #NUM2.1">
    <label>**</label>
```

```
  </iNode>
  <iNode  xml:id="EXP3"  parent="#PLU1"  children="#VARB1 #NUM2.2">
    <label>**</label>
  </iNode>
  <iNode  xml:id="PLU2"  parent="#EXP1"  children="#VARA2 #VARB2">
    <label>+</label>
  </iNode>
  <leaf  xml:id="VARA1"  parent="#EXP2">
    <label>a</label>
  </leaf>
  <leaf  xml:id="NUM2.1"  parent="#EXP2">
    <label>2</label>
  </leaf>
  <leaf  xml:id="VARB1"  parent="#EXP3">
    <label>b</label>
  </leaf>
  <leaf  xml:id="NUM2.2"  parent="#EXP3">
    <label>2</label>
  </leaf>
  <leaf  xml:id="VARA2"  parent="#PLU2">
    <label>a</label>
  </leaf>
  <leaf  xml:id="VARB2"  parent="#PLU2">
    <label>b</label>
  </leaf>
  <leaf  xml:id="NUM2.3"  parent="#EXP1">
    <label>2</label>
  </leaf>
</tree>
```

In this encoding, the **arity** attribute represents the arity of the tree, which is the greatest value of the **outDegree** attribute for any of the nodes in the tree. If, as in this case, **arity** is **2**, we say that the tree is a binary tree.

Since the left-to-right (or top-to-bottom!) order of the children of the two + nodes does not affect the arithmetic result in this case, we could represent in this tree all of the arithmetically equivalent formulas involving its leaves, by specifying the attribute **ord** as **false** on those two <iNode> elements, the attribute **ord** as **true** on the <root> and other <iNode> elements, and the attribute **ord** as **partial** on the <tree> element, as follows.

```
<tree
   n="ex2"
   ord="partial"
   arity="2"
   order="13">
  <root  xml:id="divi1"  ord="true"  children="#plu1 #exp1">
    <label>/</label>
  </root>
  <iNode
     xml:id="plu1"
     ord="false"
     parent="#divi1"
     children="#exp2 #exp3">
    <label>+</label>
  </iNode>
```

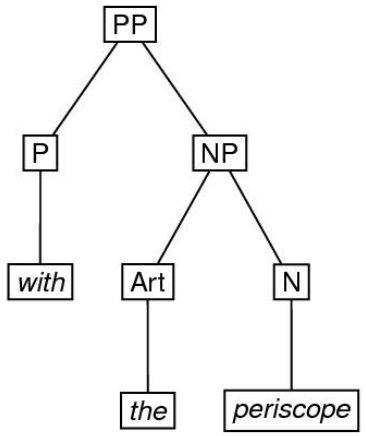
```
  <iNode
     xml:id="exp1"
     ord="true"
     parent="#divi1"
     children="#plu2 #num2.3">
    <label>**</label>
  </iNode>
  <iNode
     xml:id="exp2"
     ord="true"
     parent="#plu1"
     children="#vara1 #num2.1">
    <label>**</label>
  </iNode>
  <iNode
     xml:id="exp3"
     ord="true"
     parent="#plu1"
     children="#varb1 #num2.2">
    <label>**</label>
  </iNode>
  <iNode
     xml:id="plu2"
     ord="false"
     parent="#exp1"
     children="#vara2 #varb2">
    <label>+</label>
  </iNode>
  <leaf  xml:id="vara1"  parent="#exp2">
    <label>a</label>
  </leaf>
  <leaf  xml:id="num2.1"  parent="#exp2">
    <label>2</label>
  </leaf>
  <leaf  xml:id="varb1"  parent="#exp3">
    <label>b</label>
  </leaf>
  <leaf  xml:id="num2.2"  parent="#exp3">
    <label>2</label>
  </leaf>
  <leaf  xml:id="vara2"  parent="#plu2">
    <label>a</label>
  </leaf>
  <leaf  xml:id="varb2"  parent="#plu2">
    <label>b</label>
  </leaf>
  <leaf  xml:id="num2.3"  parent="#exp1">
    <label>2</label>
  </leaf>
</tree>
```

This encoding represents all of the following:

- `((a**2)+(b**2))/((a+b)**2)`
- `((b**2)+(a**2))/((a+b)**2)`
- `((a**2)+(b**2))/((b+a)**2)`

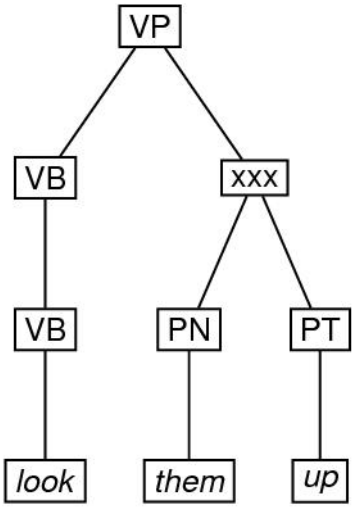
- `((b**2)+(a**2))/((a+b)**2)`

Linguistic phrase structure is very commonly represented by trees. Here is an example of phrase structure represented by an ordered tree with its root at the top, and a possible encoding.




```
<tree
   n="ex3"
   ord="true"
   arity="2"
   order="8">
  <root  xml:id="GD-PP1"  children="#GD-P1 #GD-NP1">
    <label>PP</label>
  </root>
  <iNode  xml:id="GD-P1"  parent="#GD-PP1"  children="#GD-WITH1">
    <label>P</label>
  </iNode>
  <leaf  xml:id="GD-WITH1"  parent="#GD-P1">
    <label>with</label>
  </leaf>
  <iNode  xml:id="GD-NP1"  parent="#GD-PP1"  children="#GD-THE1 #GD-PERI1">
    <label>NP</label>
  </iNode>
  <iNode  xml:id="GD-ART1"  parent="#GD-NP1"  children="#GD-THE1">
    <label>Art</label>
  </iNode>
  <leaf  xml:id="GD-THE1"  parent="#GD-ART1">
    <label>the</label>
  </leaf>
  <iNode  xml:id="GD-N1"  parent="#GD-NP1"  children="#GD-PERI1">
    <label>N</label>
  </iNode>
  <leaf  xml:id="GD-PERI1"  parent="#GD-N1">
    <label>periscope</label>
  </leaf>
</tree>
```

Finally, here is an example of an ordered tree, in which a particular node which ordinarily would precede another is specified as following it. In the drawing, the xxx symbol indicates that the arc from VB to PT crosses the arc from VP to PN.




```
<tree
   n="ex4"
   arity="2"
   order="8"
   ord="true">
  <leaf  xml:id="GD-LOOK1"  parent="#GD-VB2">
    <label>look</label>
  </leaf>
  <leaf  xml:id="GD-THEM1"  parent="#GD-PN1">
    <label>them</label>
  </leaf>
  <leaf  xml:id="GD-UP1"  parent="#GD-PT1">
    <label>up</label>
  </leaf>
  <iNode  xml:id="GD-VB2"  parent="#GD-VB1"  children="#GD-LOOK1">
    <label>VB</label>
  </iNode>
  <iNode  xml:id="GD-PN1"  parent="#GD-VP1"  children="#GD-THEM1">
    <label>PN</label>
  </iNode>
  <iNode
     xml:id="GD-PT1"
     parent="#GD-VB1"
     children="#GD-UP1"
     follow="#GD-PN1">
    <label>PT</label>
  </iNode>
  <iNode  xml:id="GD-VB1"  parent="#GD-VP1"  children="#GD-VB2 #GD-PT1">
    <label>VB</label>
  </iNode>
  <root  xml:id="GD-VP1"  children="#GD-VB1 #GD-PN1">
    <label>VP</label>
  </root>
</tree>
```

19.3 Another Tree Notation

In this section, we present an alternative to the method of representing the structure of ordered rooted trees given in section *19.2. Trees*, which is based on the observation that any node of such a tree can be thought of

as the root of the subtree that it dominates. Thus subtrees can be thought of as the same type as the trees they are embedded in, hence the designation <eTree>, for embedding tree. Whereas in a <tree> the relationship among the parts is indicated by the **children** attribute, and by the names of the elements <root>, <iNode>, and <leaf>, the relationship among the parts of an <eTree> is indicated simply by the arrangement of their content. However, we have chosen to enable encoders to distinguish the terminal elements of an <eTree> by means of the empty <eLeaf> element, though its use is not required; the <eTree> element can also be used to identify the terminal nodes of <eTree> elements. We also provide a <triangle> element, which can be thought of as an underspecified <eTree>, i.e. an <eTree> in which certain information has been left out. In addition, we provide a <forest> element, which consists of one or more <tree>, <eTree>, or <triangle> elements, and a <forestGrp> element, which consists of one or more <forest> elements. The elements used for the encoding of embedding trees and the units containing them have the following descriptions and attributes.

<eTree> (embedding tree) provides an alternative to tree element for representing ordered rooted tree structures.

@**value** provides the value of an embedding tree, which is a feature structure or other analytic element.

<triangle> (underspecified embedding tree, so called because of its characteristic shape when drawn) Provides for an underspecified eTree, that is, an eTree with information left out.

@**value** provides the value of a triangle, which is the identifier of a feature structure or other analytic element.

<eLeaf> (leaf or terminal node of an embedding tree) provides explicitly for a leaf of an embedding tree, which may also be encoded with the eTree element.

@**value** provides the value of an embedding leaf, which is a feature structure or other analytic element.

<forest> provides for groups of rooted trees.

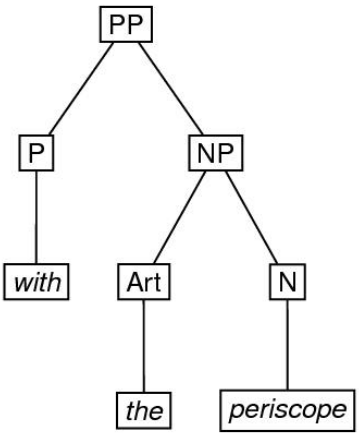
@**type** identifies the type of the forest.

<forestGrp> (forest group) provides for groups of forests.

@**type** identifies the type of the forest group.

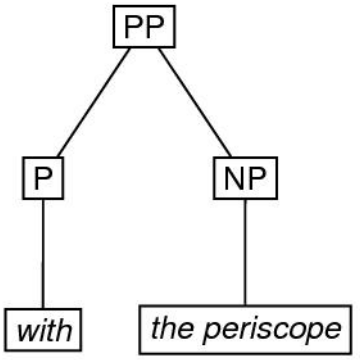
Like the <root>, <iNode>, and <leaf> of a <tree>, the <eTree>, <triangle> and <eLeaf> elements may also have **value** attributes and <label> children.

To illustrate the use of the <eTree> and <eLeaf> elements, here is an encoding of the second example in section *19.2. Trees*, repeated here for convenience.



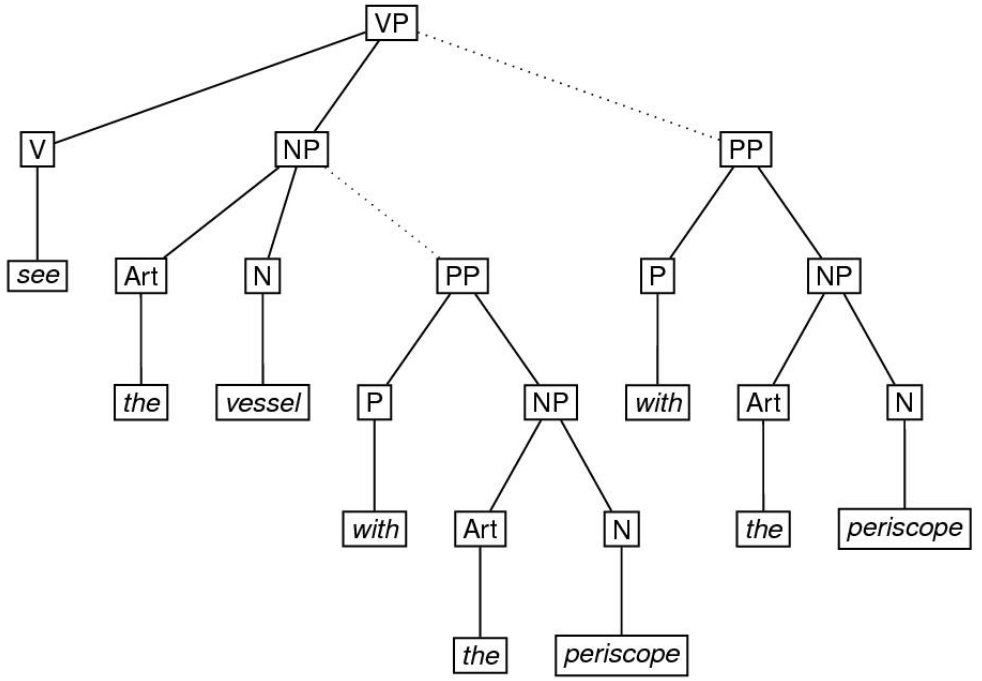
```
<eTree  n="ex1">
  <label>PP</label>
  <eTree>
    <label>P</label>
    <eLeaf>
      <label>with</label>
    </eLeaf>
  </eTree>
  <eTree>
    <label>NP</label>
    <eTree>
      <label>Art</label>
      <eLeaf>
        <label>the</label>
      </eLeaf>
    </eTree>
    <eTree>
      <label>N</label>
      <eLeaf>
        <label>periscope</label>
      </eLeaf>
    </eTree>
  </eTree>
</eTree>
```

Next, we provide an encoding, using the <triangle> element, in which the internal structure of the <eTree> labeled NP is omitted.




```
<eTree  n="ex2">
  <label>PP</label>
  <eTree>
    <label>P</label>
    <eLeaf>
      <label>with</label>
    </eLeaf>
  </eTree>
  <triangle>
    <label>NP</label>
    <eLeaf>
      <label>the periscope</label>
    </eLeaf>
  </triangle>
</eTree>
```

Ambiguity involving alternative tree structures associated with the same terminal sequence can be encoded relatively conveniently using a combination of the **exclude** and **copyOf** attributes described in sections *16.8. Alternation* and *16.6. Identical Elements and Virtual Copies*. In the simplest case, an <eTree> may be part of the content of exactly one of two different <eTree> elements. To mark it up, the embedded <eTree> may be fully specified within one of the embedding <eTree> elements to which it may belong, and a virtual copy, specified by the **copyOf** attribute, may appear on the other. In addition, each of the embedded elements in question is specified as excluding the other, using the **exclude** attribute. To illustrate, consider the English phrase *see the vessel with the periscope*, which may be considered to be structurally ambiguous, depending on whether the phrase *with the periscope* is a modifier of the phrase *the vessel* or a modifier of the phrase *see the vessel*. This ambiguity is indicated in the sketch of the ambiguous tree by means of the dotted-line arcs. The markup using the **copyOf** and **exclude** attributes follows the sketch.



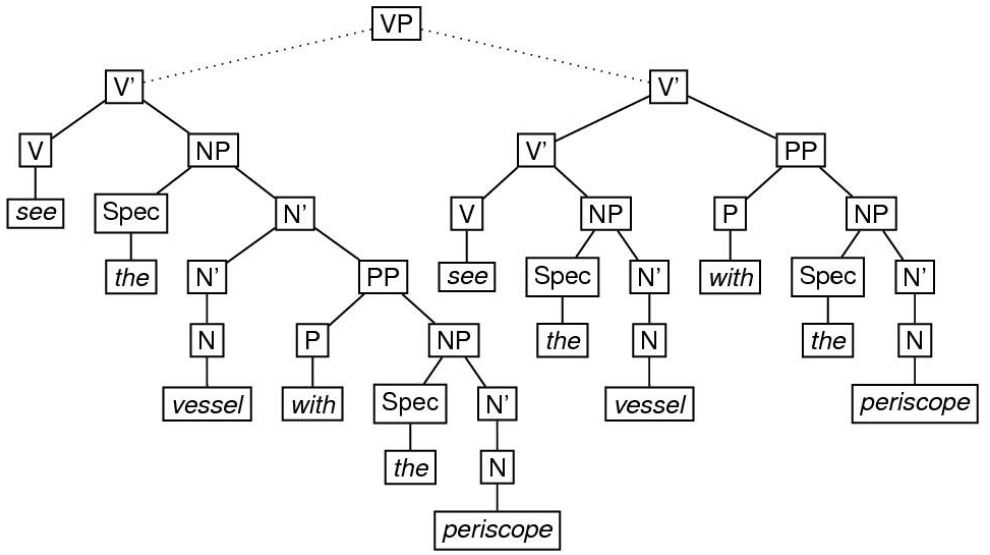

```
<eTree  n="ex3">
  <label>VP</label>
  <eTree>
    <label>V</label>
    <eLeaf>
      <label>see</label>
    </eLeaf>
  </eTree>
  <eTree>
    <label>NP</label>
    <eTree>
      <label>Art</label>
      <eLeaf>
        <label>the</label>
      </eLeaf>
    </eTree>
    <eTree>
      <label>N</label>
      <eLeaf>
```

```
      <label>vessel</label>
    </eLeaf>
  </eTree>
  <eTree xml:id="GD-PPA" exclude="#GD-PPB">
    <label>PP</label>
    <eTree>
      <label>P</label>
      <eLeaf>
        <label>with</label>
      </eLeaf>
    </eTree>
    <eTree>
      <label>NP</label>
      <eTree>
        <label>Art</label>
        <eLeaf>
          <label>the</label>
        </eLeaf>
      </eTree>
      <eTree>
        <label>N</label>
        <eLeaf>
          <label>periscope</label>
        </eLeaf>
      </eTree>
    </eTree>
  </eTree>
 <eTree xml:id="GD-PPB" copyOf="#GD-PPA" exclude="#GD-PPA">
   <label>PP</label>
 </eTree>
</eTree>
```

To indicate that one of the alternatives is selected, one may specify the **select** attribute on the highest <eTree> as either **#GD-PPA** or **#GD-PPB**; see section *16.8. Alternation*.

Depending on the grammar one uses to associate structures with examples like *see the man with the periscope*, the representations may be more complicated than this. For example, adopting a version of the X-bar theory of phrase structure originated by Jackendoff,[3] the attachment of a modifier may require the creation of an intermediate node which is not required when the attachment is not made, as shown in the following diagram. A possible encoding of this ambiguous structure immediately follows the diagram.

[3] Jackendoff (1977)

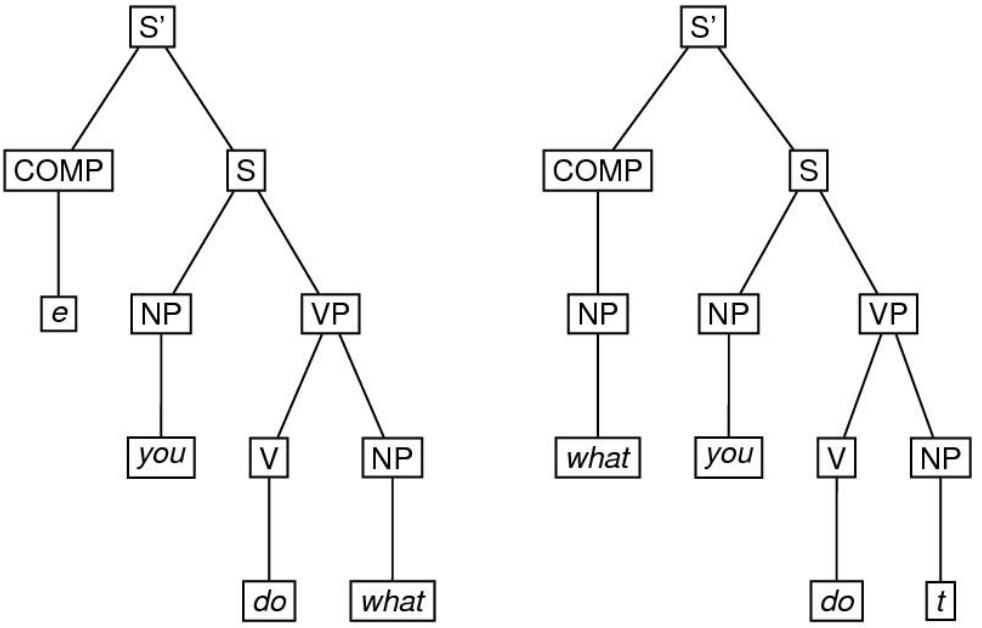



```
<eTree n="ex4">
  <label>VP</label>
  <eTree xml:id="VBARA" exclude="#VBARB">
    <label>V'</label>
    <eTree xml:id="VA">
      <label>V</label>
      <eLeaf>
        <label>see</label>
      </eLeaf>
    </eTree>
    <eTree>
      <label>NP</label>
      <eTree xml:id="SPEC1A">
        <label>Spec</label>
        <eLeaf>
          <label>the</label>
        </eLeaf>
      </eTree>
      <eTree>
        <label>N'</label>
        <eTree xml:id="NBAR2A">
          <label>N'</label>
          <eTree>
            <label>N</label>
            <eLeaf>
              <label>vessel</label>
            </eLeaf>
          </eTree>
        </eTree>
        <eTree xml:id="PPA1">
          <label>PP</label>
          <eTree>
            <label>P</label>
            <eLeaf>
              <label>with</label>
            </eLeaf>
          </eTree>
```

```
        <eTree>
          <label>NP</label>
          <eTree>
            <label>Spec</label>
            <eLeaf>
              <label>the</label>
            </eLeaf>
          </eTree>
          <eTree>
            <label>N'</label>
            <eTree>
              <label>N</label>
              <eLeaf>
                <label>periscope</label>
              </eLeaf>
            </eTree>
          </eTree>
        </eTree>
      </eTree>
    </eTree>
  </eTree>
  <eTree xml:id="VBARB" exclude="#VBARA">
    <label>V'</label>
    <eTree>
      <label>V'</label>
      <eTree xml:id="VB" copyOf="#VA">
        <label>V</label>
      </eTree>
      <eTree>
        <label>NP</label>
        <eTree xml:id="SPEC1B" copyOf="#SPEC1A">
          <label>Spec</label>
        </eTree>
        <eTree xml:id="NBAR2B" copyOf="#NBAR2A">
          <label>N'</label>
        </eTree>
      </eTree>
    </eTree>
    <eTree xml:id="PPB" copyOf="#PPA1">
      <label>PP</label>
    </eTree>
  </eTree>
</eTree>
```

A derivation in a generative grammar is often thought of as a set of trees. To encode such a derivation, one may use the <forest> element, in which the trees may be marked up using the <tree>, the <eTree>, or the <triangle> element. The **type** attribute may be used to specify what kind of derivation it is. Here is an example of a two-tree forest, involving application of the 'wh-movement' transformation in the derivation of *what you do* (as in *this is what you do*) from the underlying *you do what*.[4]

[4] The symbols e and t denote special theoretical constructs (empty category and trace respectively), which need not concern us here.




```
<forest n="ex5" type="derivation-syntactic">
  <eTree n="Stage 1" xml:id="S1SBAR">
    <label>S'</label>
    <eTree xml:id="S1COMP">
      <label>COMP</label>
      <eLeaf xml:id="S1E">
        <label>e</label>
      </eLeaf>
    </eTree>
    <eTree xml:id="S1S">
      <label>S</label>
      <eTree xml:id="S1NP1">
        <label>NP</label>
        <eLeaf>
          <label>you</label>
        </eLeaf>
      </eTree>
      <eTree xml:id="S1VP">
        <label>VP</label>
        <eTree xml:id="S1V">
          <label>V</label>
          <eLeaf>
            <label>do</label>
          </eLeaf>
        </eTree>
        <eTree xml:id="S1NP2">
          <label>NP</label>
          <eLeaf xml:id="S1WH">
            <label>what</label>
          </eLeaf>
        </eTree>
      </eTree>
    </eTree>
  </eTree>
  <eTree n="Stage 2" xml:id="S2SBAR" corresp="#S1SBAR">
```

```
    <label>S'</label>
    <eTree xml:id="S2COMP" corresp="#S1COMP">
      <label>COMP</label>
      <eTree copyOf="#S1NP2" corresp="#S1E">
        <label>NP</label>
      </eTree>
    </eTree>
    <eTree xml:id="S2S" corresp="#S1S">
      <label>S</label>
      <eTree xml:id="S2NP1" copyOf="#S1NP1">
        <label>NP</label>
      </eTree>
      <eTree xml:id="S2VP" corresp="#S1VP">
        <label>VP</label>
        <eTree xml:id="S2V" copyOf="#S1V">
          <label>V</label>
        </eTree>
        <eTree xml:id="S2NP2" corresp="#S1NP2">
          <label>NP</label>
          <eLeaf corresp="#S1WH">
            <label>t</label>
          </eLeaf>
        </eTree>
      </eTree>
    </eTree>
  </eTree>
</forest>
```

In this markup, we have used **copyOf** attributes to provide virtual copies of elements in the tree representing the second stage of the derivation that also occur in the first stage, and the **corresp** attribute (see section *16.4. Correspondence and Alignment*) to link those elements in the second stage with corresponding elements in the first stage that are not copies of them.

If a group of forests (e.g. a full grammatical derivation including syntactic, semantic, and phonological subderivations) is to be articulated, the grouping element <forestGrp> may be used.

19.4 Representing Textual Transmission

A stemma codicum (sometimes called just stemma) is a tree-like graphic structure that has become traditional in manuscript studies for representing textual transmission. Consider the following hypothetical stemma:

The nodes in this stemma represent manuscripts; each has a label (a letter) which identifies it and also distinguishes whether the manuscript is extant, lost, or hypothetical. Extant manuscripts are identified by uppercase Latin letters or words beginning with uppercase Latin letters, e.g., **L**, shown as aqua in this example; manuscripts no longer existing, but providing readings which are attested e.g. by note or copy made before their disappearance, are identified by lowercase Latin letters, e.g., **t**, shown as magenta in this example; hypothetical stages in the textual transmission, which do not necessarily correspond to real manuscripts, are given lowercase Greek letters, e.g., **α** and shown as gold in this example. The stemma shown above thus suggests that (on the basis of similarities in the readings of the extant and lost manuscripts) **L** and **t** share textual material that is not shared with other manuscripts (represented in this case by **δ**) even though no physical manuscript attesting this stage in the textual transmission has ever been identified.

Manuscripts are copied from other manuscripts. The preceding stemma represents the hypothesis that all manuscripts go back to a common ancestor (**α**), that the tradition split after that stage into two (**β** and **γ**), etc. Descent by copying is indicated with a solid line. According to this model, **α** is the earliest common

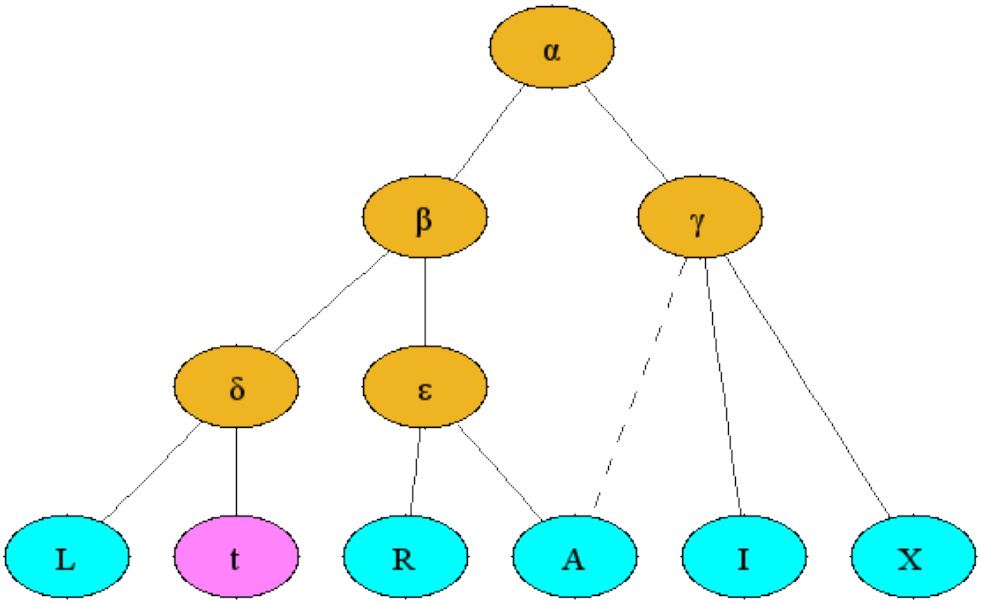


Figure 19.1: Example stemma

hypothetical stage that can be reconstructed, and all nodes below **α** have a single parent, that is, were copied from a single other stage in the tradition.

This familiar tree model is complicated because manuscripts sometimes show the influence of more than one ancestor. They may have been produced by a scribe who checked the text in one manuscript of the same work whilst copying from another, or perhaps made changes from his memory of a slightly different version of the text that he had read elsewhere. Alternatively, perhaps scribe A copied a manuscript from one source, scribe B made changes in it in the margins or between the lines (either by consulting another source directly or from memory), and another scribe then copied that manuscript, incorporating the changes into the body. Whatever the specific scenario, it is not uncommon for a manuscript to be based primarily on one source, but to incorporate features of another branch of the tradition. This mixed result is called contamination, and it is reflected in a stemma by a dotted line. Thus, the example above asserts that **A** is copied within the **ε** tradition, but is also contaminated from the **γ** tradition.

The utility of a stemma as a visualization tool is inversely proportional to the degree of contamination in the manuscript tradition. A tradition completely without contamination (called a closed tradition) yields a classic tree, easily represented graphically by a stemma. An open tradition, with substantial contamination, yields a spaghetti-like stemma characterized by crossing dotted lines, which is both difficult to read and not very informative.

The <eTree> element introduced in this chapter can be used to represent a closed tradition in a straightforward manner. Each non-terminal node is represented by a typed <eTree> element and each terminal node by an <eLeaf>. A <label> element provides a way of identifying each node, complementary to the global attributes **n** and **xml:id** attributes. For example, the closed part of the tradition headed by the label δ may be encoded as follows:

```
<eTree type="hypothetical">
  <label>δ</label>
  <eLeaf type="extant">
    <label>L</label>
  </eLeaf>
```

```
  <eLeaf type="lost">
    <label>t</label>
  </eLeaf>
</eTree>
```

To complete this representation, we need to show that the node labelled A is not derived solely from its parent node (labelled ε) but also demonstrates contamination from the node labelled γ. The easiest way to accomplish this is to include an appropriately-typed <ptr> element within the node in question, the **target** of which points to the node labelled γ. This requires that this latter node be supplied with a value for its **xml:id** attribute. The complete representation is thus:

```
<eTree type="hypothetical">
  <label>α</label>
  <eTree type="hypothetical">
    <label>β</label>
    <eTree type="hypothetical">
      <label>δ</label>
      <eLeaf type="extant">
        <label>L</label>
      </eLeaf>
      <eLeaf type="lost">
        <label>t</label>
      </eLeaf>
    </eTree>
    <eTree type="hypothetical">
      <label>ε</label>
      <eLeaf type="extant">
        <label>R</label>
      </eLeaf>
      <eLeaf type="extant">
        <label>A</label>
        <ptr type="contamination" target="#gamma"/>
      </eLeaf>
    </eTree>
  </eTree>
  <eTree xml:id="gamma" type="hypothetical">
    <label>γ</label>
    <eLeaf type="extant">
      <label>I</label>
    </eLeaf>
    <eLeaf type="extant">
      <label>X</label>
    </eLeaf>
  </eTree>
</eTree>
```

In any substantial codicological project, it is likely that significantly more data will be required about the individual witnesses than indicated in the simple structures above. These Guidelines provide a rich variety of additional elements for representing such information: see in particular chapters *10. Manuscript Description*, *11. Representation of Primary Sources*, and *12. Critical Apparatus*.

19.5 Module for Graphs, Networks, and Trees

The module described in this chapter makes available the following components:

Module nets: Graphs, networks, and trees

- Elements defined: arc eLeaf eTree forest forestGrp graph iNode leaf node root tree triangle

The selection and combination of modules to form a TEI schema is described in *1.2. Defining a TEI Schema*.

Chapter 20

Non-hierarchical Structures

XML employs a strongly hierarchical document model. At various points, these Guidelines discuss problems that arise when using XML to encode textual features that either do not naturally lend themselves to representation in a strictly hierarchical form or conflict with other hierarchies represented in the markup. Examples of such situations include:

- Conflict between the hierarchy established by the physical structure of a document (e.g., volume, page, column, line) and its rhetorical or linguistic structure (e.g., chapters, paragraphs, sentences, acts, scenes, etc.)
- Conflict between a verse text's metrical structure (e.g., its arrangement in stanzas and metrical lines) and its rhetorical or linguistic structure (e.g., phrases, sentences, and, for plays, acts, scenes, and speeches).
- Conflict between metrical, rhetorical, or linguistic structure and the representation of direct speech, especially if the quoted speech is interrupted by other elements (e.g., What, she asked, was that all about) or crosses metrical, rhetorical, or linguistic boundaries.
- Conflict between different analytical views or descriptions of a text or document, e.g., markup intended to encode diplomatic information about a word's appearance in a manuscript with markup intended to describe its morphology or pronunciation.

Non-nesting information poses fundamental problems for any XML-based encoding scheme, and it must be stated at the outset that no current solution combines all the desirable attributes of formal simplicity, capacity to represent all occurring or imaginable kinds of structures, suitability for formal or mechanical validation. The representation of non-hierarchical information is thus necessarily a matter of trade-offs among various sets of advantages and disadvantages.

These Guidelines support several methods for handling non-hierarchical information:

- redundant encoding of information in multiple forms (discussed in *20.1. Multiple Encodings of the Same Information*)
- the use of empty elements to delimit the boundaries of a non-nesting structure (discussed in *20.2. Boundary Marking with Empty Elements*)
- the division of a logically single non-nesting element into segments that nest properly in their immediate hierarchical context but can also be reconstituted virtually across these hierarchical boundaries (discussed *20.3. Fragmentation and Reconstitution of Virtual Elements*)
- stand-off markup: the annotation of information by pointing at it, rather than by placing XML tags within it (discussed in *20.4. Stand-off Markup*)

Some of these methods can be used in TEI Conformant or Conformable documents. Others require extension.

In the sections which follow these techniques are described and their advantages and disadvantages are briefly discussed. The various solutions to the problem will be exemplified using extracts from two poems. The first is the opening quatrain from William Wordsworth's 'Scorn not the sonnet':

> Scorn not the sonnet; critic, you have frowned,
> Mindless of its just honours; with this key
> Shakespeare unlocked his heart; the melody
> Of this small lute gave ease to Petrarch's wound.

The second example is the third stanza from the fourth section of Robert Pinsky's 'Essay on Psychiatrists':

> Catholic woman of twenty-seven with five children
> And a first-rate body—pointed her finger
> at the back of one certain man and asked me,
> "Is that guy a psychiatrist?" and by god he was! "Yes,"
> She said, "He *looks* like a psychiatrist."
> Grown quiet, I looked at his pink back, and thought.

These two texts can be analysed in various ways. The first, which we might describe as the 'Metrical View', encodes the text according to its metrical features: line divisions (as here), stanzas or cantos in larger poems, and perhaps prosodic features like stress or syllable patterns, alliteration, or rhyme. A second view, which we might describe as the 'Grammatical', encodes linguistic and rhetorical features: phonemes, morphemes, words, phrases, clauses, and sentences. A third view, the 'Dialogic', might concentrate on narrative voice: distinguishing between the narrator and their interlocutors and identifying individual segments as direct quotations. In our examples, we will restrict ourselves to relatively simple conflicts: for the Metrical View we will encode only metrical lines and line groups; for the Grammatical View we will restrict ourselves to encoding sentences; and for the Dialogic View, we only will distinguish direct quotation from other narration.

20.1 Multiple Encodings of the Same Information

Conceptually, the simplest method of disentangling two (or more) conflicting hierarchical views of the same information is to encode it twice (or more), each time capturing a single view.

Thus, for example, the Metrical View of 'Scorn not the sonnet' might be encoded as follows, using the <l> element to encode each metrical line:

```
<l>Scorn not the sonnet; critic, you have frowned,</l>
<l>Mindless of its just honours; with this key</l>
<l>Shakespeare unlocked his heart; the melody</l>
<l>Of this small lute gave ease to Petrarch's wound.</l>
```

Source: [210]

The Grammatical View would be encoded by taking the same text and replacing the metrical markup with information about its sentence structure:

```
<p>
  <seg>Scorn not the sonnet;</seg>
  <seg>critic, you have frowned, Mindless of its just honours;</seg>
  <seg>with this key Shakespeare unlocked his heart;</seg>
  <seg>the melody Of this small lute gave ease to Petrarch's wound.</seg>
</p>
```

Source: [210]

Likewise, the more complex passage from Pinsky could be encoded in three different ways to reflect the different metrical, grammatical, and dialogic views of its text:

```
<lg>
  <l>Catholic woman of twenty-seven with five children</l>
  <l>And a first-rate body—pointed her finger</l>
  <l>at the back of one certain man and asked me,</l>
  <l>"Is that guy a psychiatrist?" and by god he was! "Yes,"</l>
  <l>She said, "He <emph>looks</emph> like a psychiatrist."</l>
  <l>Grown quiet, I looked at his pink back, and thought.</l>
</lg>
```

Source: [156]

```
<p>
  <seg>Catholic woman of twenty-seven with five children And a first-rate
     body—pointed her finger at the back of one certain man and asked me,
     "Is that guy a psychiatrist?" and by god he was!</seg>
</p>
<p>
  <seg>"Yes," She said, "He <emph>looks</emph> like a psychiatrist."</seg>
</p>
<p>
  <seg>Grown quiet, I looked at his pink back, and thought.</seg>
</p>
```

Source: [156]

```
<ab>Catholic woman of twenty-seven with five children And a first-rate
body—pointed her finger at the back of one certain man and asked me,
<said>Is that guy a psychiatrist?</said> and by god he was!
<said>Yes,</said> She said, <said>He <emph>looks</emph> like a
    psychiatrist.</said> Grown quiet, I looked at his pink back, and
thought.</ab>
```

Source: [156]

This method is TEI Conformant. Its advantages are that each way of looking at the information is explicitly represented in the data and that the individual views are simple to process. The disadvantages are that the method requires the maintenance of multiple copies of identical textual content (an invitation to inconsistency) and that there is no explicit indication that the various views, which might be in separate files, are related to each other: it might prove difficult to combine the views or access information from one view while processing the file that contains the encoding of another.[1]

20.2 Boundary Marking with Empty Elements

A second method for accommodating non-hierarchical objects in an XML document involves marking the start and end points of the non-nesting material. This prevents textual features that fall outside the privileged hierarchy from invalidating the document while identifying their beginnings and ends for further processing.

[1] It has been shown, however, that it is possible to relate the different annotations in an indirect way: if the textual content of the annotations is identical, the very text can serve as a means for linking the different annotations, as described in Witt (2002).

The disadvantage of this method is that no single XML element represents the non-nesting material and, as a result, processing with XML technologies is significantly more difficult.

The empty elements used at each end are called segment-boundary elements or segment-boundary delimiters. There are several variations on this method of encoding.

For some common structural features, the TEI provides milestone elements that can be used to mark the beginning of a textual feature. These include <lb>, <pb>, <cb>, <handShift>, and the generic <milestone>. Using <lb>, for example, it is possible to indicate both the physical lineation of a poem on the page and its grammatical division into sentences:

```
<p>
  <seg>
    <lb  n="1"/>Scorn not the sonnet;</seg>; <seg>critic, you have
     frowned, <lb  n="2"/>Mindless of its just honours;</seg>
  <seg>with this key <lb  n="3"/>Shakespeare unlocked his heart;</seg>
  <seg>the melody <lb  n="4"/>Of this small lute gave ease to Petrarch's
     wound.</seg>
</p>
```

Source: [210]

The use of these elements is by definition TEI Conformant. Care should be taken, however, that the meaning of the milestone elements is preserved: semantically, for example, <lb> is used to mark the start of a new (typographical) line. While in much modern poetry, typographical and metrical line divisions correspond, <lb> does not itself make a metrical claim: in encoding verse from sources, such as Old English manuscripts, where physical line breaks are not used to indicate metrical lineation, the correspondence would break down entirely.

The segment boundaries also may be delimited by the generic <anchor> element. Attributes can then be used to indicate the type of feature being delimited and whether a given instance opens or closes the feature.

```
<l>
  <anchor  subtype="sentenceStart"  type="delimiter"/>
Scorn not the sonnet;
<anchor  subtype="sentenceEnd"  type="delimiter"/>
  <anchor  subtype="sentenceStart"  type="delimiter"/> critic, you have frowned,
</l>
<l>Mindless of its just honours; <anchor  subtype="sentenceEnd"  type="delimiter"/>
  <anchor  subtype="sentenceStart"  type="delimiter"/> with this key</l>
<l>Shakespeare unlocked his heart; <anchor  subtype="sentenceEnd"  type="delimiter"/>
  <anchor  subtype="sentenceStart"  type="delimiter"/> the melody</l>
<l>Of this small lute gave ease to Petrarch's wound. <anchor  subtype="sentenceEnd"  type="delimiter"/>
</l>
```

Source: [210]

This method is TEI Conformant.

Another approach is to design custom elements that provide richer information about the feature being delimited or its boundaries. This information can be included as attribute values or as part of the element name itself: e.g., **<boundaryStart element="sentence"/>**... **<boundaryEnd element="sentence"/>**, **<sentenceBoundary position="start"/>**... **<sentenceBoundary position="end"/>**, or **<sentenceBoundaryStart/>**... **<sentenceBoundaryEnd/>**:

```
<l
   xmlns:n="http://www.example.org/ns/nonTEI">
  <sentenceBoundaryStart xmlns="http://www.example.org/ns/nonTEI"
  />Scorn not the sonnet;
<sentenceBoundaryEnd xmlns="http://www.example.org/ns/nonTEI"
  />
  <sentenceBoundaryStart xmlns="http://www.example.org/ns/nonTEI"
  />critic, you have frowned,
</l>
<l>Mindless of its just honours; <sentenceBoundaryEnd xmlns="http://www.example.org/ns/nonTEI"
  />
  <sentenceBoundaryStart xmlns="http://www.example.org/ns/nonTEI"
  />with this key</l>
<l>Shakespeare unlocked his heart; <sentenceBoundaryEnd xmlns="http://www.example.org/ns/nonTEI"
  />
  <sentenceBoundaryStart xmlns="http://www.example.org/ns/nonTEI"
  />the melody</l>
<l>Of    this    small    lute    gave    ease    to    Petrarch's    wound.          <sentenceBoundaryEnd
xmlns="http://www.example.org/ns/nonTEI"
  />
</l>
```

Source: [210]

If the custom elements can be replaced by TEI elements and attributes without loss of information, this method is TEI Conformable (see *23.3. Conformance*); if the custom elements introduce information or distinctions that cannot be captured using standard TEI elements, the method is an extension.

Finally, elements that are normally used to encode nesting textual features (e.g., <said>, <seg>, <l>, etc.) can be adapted so that they serve as empty segment boundary delimiters when the features they encode cross-hierarchical boundaries. Additional attributes (**sID** and **eID** in the example below) are added to these elements in order to allow the unambiguous correlation of start and end points. This method has been introduced in the markup literature under various names, including Trojan milestones, HORSE markup, CLIX, and COLT. It is described in detail by DeRose (2004)):

```
<lg
   xmlns:hr="http://www.example.org/ns/nonTEI">
  <l>
    <seg>Scorn not the sonnet;</seg>
    <s xmlns="http://www.example.org/ns/nonTEI"
      sID="s02"/>critic, you have frowned, </l>
  <l>Mindless of its just honours; <s xmlns="http://www.example.org/ns/nonTEI"
      eID="s02"/>
    <s xmlns="http://www.example.org/ns/nonTEI"
      sID="s03"/>with this key </l>
  <l>Shakespeare unlocked his heart; <s xmlns="http://www.example.org/ns/nonTEI"
      eID="s03"/>
    <s xmlns="http://www.example.org/ns/nonTEI"
      sID="s04"/>the melody </l>
  <l>Of this small lute gave ease to Petrarch's wound. <s xmlns="http://www.example.org/ns/nonTEI"
      eID="s04"/>
  </l>
</lg>
```

Source: [210]

Depending on how the modifications are carried out, this method may be TEI Conformable, represent an extension of the TEI, or produce a non-conformant document.

- The method is TEI Conformable if the modified elements are placed in a distinct, non-TEI-namespace (see *23.3.4. Use of the TEI Namespace*), and if the modified elements and attributes can be mapped without loss of information to existing TEI markup structures such as milestone or anchor elements automatically (see *23.3. Conformance*).
- The method represents an Extension if the modified elements are placed in a distinct, non-TEI namespace, but contain information or distinctions that cannot be algorithmically translated to existing TEI elements without loss of information (see *23.3. Conformance*).
- The method is non-conformant—and indeed strongly deprecated—if the modified elements and attributes are not placed in a distinct, non-TEI namespace (see *23.3.3. Conformance to the TEI Abstract Model*).

In each of the above examples (except the last), the relationship between the start and end delimiters (where these exist) of a given feature is implicit: it is assumed that "end" delimiters close the nearest preceding "start" delimiter, or, in the case of milestones, that the milestone marks both the end of the preceding example and the beginning of the next. Complications arise, however, when the non-nesting text overlaps with other non-nesting text of the same type, as, for example, in a grammatical analysis of the various possible interpretations of the noun phrase *fast trains and planes*. In this case, the adjective *fast* can be understood as either modifying *trains and planes* or just *trains*:

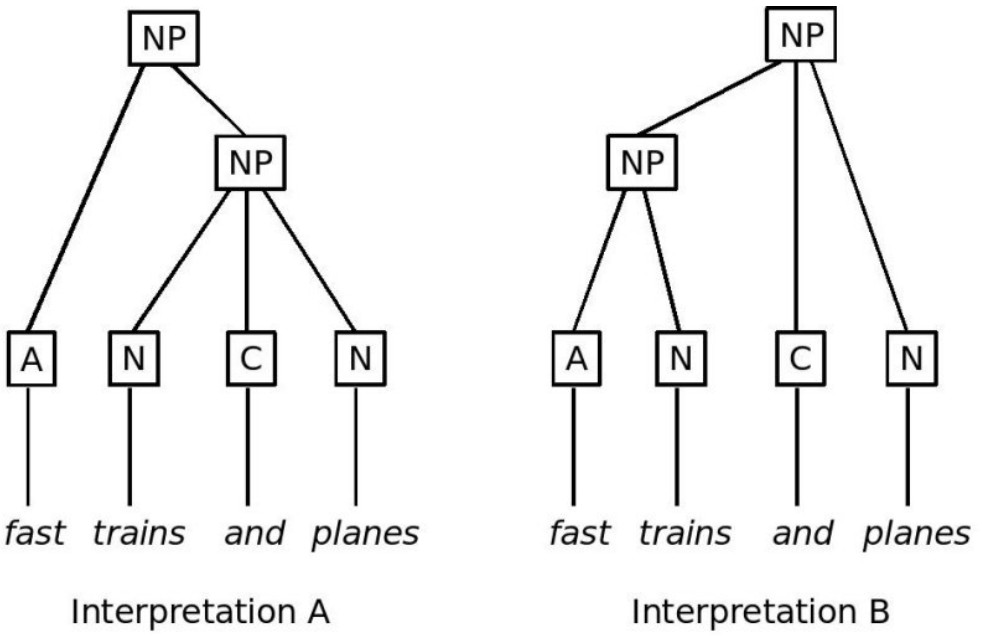


Figure 20.1: Two interpretations of the phrase *Fast trains and planes*

In order to encode the possible analyses of this phrase, an unambiguous method of associating opening and closing segment boundary delimiters is required:

```
<phr function="NP">
  <anchor type="delimiter" subtype="NPstart" xml:id="NPInterpretationB"/>
  <w function="A">Fast</w>
  <anchor type="delimiter" subtype="NPstart" xml:id="NPInterpretationA"/>
  <w function="N">trains</w>
```

```
  <anchor type="delimiter" subtype="NPend" corresp="#NPInterpretationB"/>
  <w function="C">and</w>
  <w function="N">planes</w>
  <anchor type="delimiter" subtype="NPend" corresp="#NPInterpretationA"/>
</phr>
```

Source: [124]

In this encoding, the first interpretation, in which *fast* modifies the NP *trains and planes*, the NP *trains and planes* is opened using an <anchor> tag with the **xml:id** value *NPInterpretationA* and closed with an <anchor> with the same value on **corresp**; in the second interpretation, in which *fast* forms a NP with *trains*, the NP *fast cars* is opened using an <anchor> tag with the **xml:id** value *NPInterpretationB* and closed with an <anchor> tag that has the same value on **corresp**.

Despite their advantages, segment boundary delimiters incur the disadvantage of cumbersome processing: since the elements of the analysis (e.g., the sentences in the poems, or phrases in the above example) are not uniformly represented by nodes in the document tree, they must be reconstituted by software in an ad hoc fashion, which is likely to be difficult and may be error prone.

Most important for some encoders, the method also disguises the relationship between the beginning and the ending of each logical element. This makes it impossible for standard validation software to provide the same kind of validation possible elsewhere in the encoding. When using grammar-based schema languages it is not possible to define a content model for the range limited by empty elements.[2]

20.3 Fragmentation and Reconstitution of Virtual Elements

A third method involves breaking what might be considered a single logical (but non-nesting) element into multiple smaller structural elements that fit within the dominant hierarchy but can be reconstituted virtually. For example, if a passage of direct discourse begins in the middle of one paragraph and continues for several more paragraphs, one could encode the passage as a series of <said> elements, each fitting within a <p> element. The resulting encoding is valid XML, but the text in each <said> element represents only a portion of the complete passage of direct discourse. For this reason these elements are sometimes called 'partial elements'.

In the case of our selection from Pinsky's poem, for example, the second passage of direct quotation, which crosses a line boundary and is broken up by a *She said* in the narrator's voice, can be made to fit within the hierarchy established by the metrical lineation by using two <said> elements:

```
<lg>
  <l>Catholic woman of twenty-seven with five children</l>
  <l>And a first-rate body—pointed her finger</l>
  <l>at the back of one certain man and asked me,</l>
  <l>
    <said n="quotation1">Is that guy a psychiatrist?</said> and by god he was!
  <said n="quotation2">Yes,</said>
  </l>
  <l>She said, <said n="quotation2">He <emph>looks</emph> like a
        psychiatrist.</said>
  </l>
  <l>Grown quiet, I looked at his pink back, and thought.</l>
</lg>
```

Source: [156]

[2] Grammar based schema languages (e.g., DTD, W3C Schema, and RELAX NG) are used to define markup languages (e.g., XHTML or TEI). Rule-based schema languages (e.g., Schematron) can be used to define further constraints. Such a rule-based schema language permits a sequence of certain elements between empty elements to be legitimized or prohibited.

Similarly, the sentences in our example from Wordsworth could be encoded:

```
<l>
  <seg  n="sentence1">Scorn not the sonnet;</seg>
  <seg  n="sentence2">critic, you have frowned,</seg>
</l>
<l>
  <seg  n="sentence2">Mindless of its just honours;</seg>
  <seg  n="sentence3">with this key</seg>
</l>
<l>
  <seg  n="sentence3">Shakespeare unlocked his heart;</seg>
  <seg  n="sentence4">the melody</seg>
</l>
<l>
  <seg  n="sentence4">Of this small lute gave ease to Petrarch's wound.</seg>
</l>
```

Source: [210]

There are two main problems with this type of encoding. The first is that it invariably means that the encoding will have more elements claiming to represent a feature than there are actual instances of that feature in the text. Thus, for example, the passage from 'Scorn not the sonnet' marks seven spans of text using <seg>, even though there are only four linguistic sentences in the passage.

The second problem is that it can be semantically misleading. Although they are tagged using the element for sentence, for example, very few of the textual features encoded using <seg> in this example represent actual linguistic sentences: *with this key*, for example, is a prepositional phrase, not a sentence; *Of this small lute gave ease to Petrarch's wound* is a string corresponding to no single grammatical category.

Taken together, these problems can make automatic analysis of the fragmented features difficult. An analysis that intended to count the number of sentences in Wordsworth's poem, for example, would arrive at an inflated figure if it understood the <seg> elements to represent complete rhetorical sentences; if it wanted to do an analysis of his syntax, it would not be able to assume that <seg> delimited linguistic sentences.

The technique of fragmentation is often complemented by the technique of virtual joins. Virtual joins may be used to combine objects in the text to a new hierarchy. Here is 'Scorn not the sonnet' again; this time the relationship between the parts of the fragmented sentences is indicated explicitly using the **next** and **prev** attributes described in *16.7. Aggregation*.

```
<l>
  <seg>Scorn not the sonnet;</seg>
  <seg  next="#s2b"  xml:id="s2a">critic, you have frowned,</seg>
</l>
<l>
  <seg  prev="#s2a"  xml:id="s2b">Mindless of its just honours;</seg>
  <seg  next="#s3b"  xml:id="s3a">with this key</seg>
</l>
<l>
  <seg  prev="#s3a"  xml:id="s3b">Shakespeare unlocked his heart;</seg>
  <seg  next="#s4b"  xml:id="s4a">the melody</seg>
</l>
<l>
  <seg  prev="#s4a"  xml:id="s4b">Of this small lute gave ease to Petrarch's wound.</seg>
</l>
```

Source: [210]

This method of virtually joining partial elements is sometimes called 'chaining'.

For fragments encoded using <ab>, <l>, <lg>, , or elements that belong to the att.segLike class, an even simpler mechanism for virtually joining fragments exists: the use of the **part** attribute with the value *I* (Initial), *M* (Medial), or *F* (Final) as described in *16.3. Blocks, Segments, and Anchors.* Here is the above example recoded to reflect this method:

```
<l>
  <seg>Scorn not the sonnet;</seg>
  <seg  part="I">critic, you have frowned,</seg>
</l>
<l>
  <seg  part="F">Mindless of its just honours;</seg>
  <seg  part="I">with this key</seg>
</l>
<l>
  <seg  part="F">Shakespeare unlocked his heart;</seg>
  <seg  part="I">the melody</seg>
</l>
<l>
  <seg  part="F">Of this small lute gave ease to Petrarch's wound.</seg>
</l>
```

Source: [210]

This method is TEI Conformant and simple to use. Its disadvantage is that it does not work well for cases of self-overlap, or if there are nested occurrences of the same element type, as it can become difficult to ascertain which initial, medial, or final partial element should be combined with which others or in which order. This problem becomes evident if we attempt to combine a detailed Grammatical view of the Pinsky example with its metrical encoding:

```
<lg>
  <l>
    <seg  part="I">Catholic woman of twenty-seven with five children</seg>
  </l>
  <l>
    <seg  part="M">And a first-rate body—pointed her finger</seg>
  </l>
  <l>
    <seg  part="M">at the back of one certain man and asked me,</seg>
  </l>
  <l>
    <seg  part="F">"<seg>Is that guy a psychiatrist?</seg>" and by god he was!</seg>
    <seg  part="I">"<seg  part="I">Yes,</seg>"</seg>
  </l>
  <l>
    <seg  part="F">She said, "<seg  part="F">He <emph>looks</emph> like a psychiatrist.</seg>"</seg>
  </l>
  <l>
    <seg>Grown quiet, I looked at his pink back, and thought.</seg>
  </l>
</lg>
```

Source: [156]

A third method for aggregating fragmented partial elements involves using markup that is not directly part of the encoding, e.g., the <join> element. In this method, a <join> element is used elsewhere in the document to indicate explicitly the members of the virtual element:

```
<l>
  <w  xml:id="w01">Scorn</w>
  <w  xml:id="w02">not</w>
  <w  xml:id="w03">the</w>
  <w  xml:id="w04">sonnet</w>; <w  xml:id="w05">critic</w>, <w  xml:id="w06">you</w>
  <w  xml:id="w07">have</w>
  <w  xml:id="w08">frowned</w>,
</l>
<l>
  <w  xml:id="w09">Mindless</w>
  <w  xml:id="w10">of</w>
  <w  xml:id="w11">its</w>
  <w  xml:id="w12">just</w>
  <w  xml:id="w13">honours</w>; <w  xml:id="w14">with</w>
  <w  xml:id="w15">this</w>
  <w  xml:id="w16">key</w>
</l>
<l>
  <w  xml:id="w17">Shakespeare</w>
  <w  xml:id="w18">unlocked</w>
  <w  xml:id="w19">his</w>
  <w  xml:id="w20">heart</w>; <w  xml:id="w21">the</w>
  <w  xml:id="w22">melody</w>
</l>
<l>
  <w  xml:id="w23">Of</w>
  <w  xml:id="w24">this</w>
  <w  xml:id="w25">small</w>
  <w  xml:id="w26">lute</w>
  <w  xml:id="w27">gave</w>
  <w  xml:id="w28">ease</w>
  <w  xml:id="w29">to</w>
  <w  xml:id="w30">Petrarch's</w>
  <w  xml:id="w31">wound</w>.
</l>
<!-- Elsewhere in the document -->
<p>
  <join  result="s"  scope="root"  targets="#w01 #w02 #w03 #w04"/>
  <join
     result="s"
     scope="root"
     targets="#w05 #w06 #w07 #w08 #w09 #w10 #w11 #w12 #w13"/>
  <join  result="s"  scope="root"
     targets="#w14 #w15 #w16 #w17 #w18 #w19 #w20"/>
  <join
     result="s"
     scope="root"
     targets="#w21 #w22 #w23 #w24 #w25 #w26 #w27 #w28 #w29 #w30 #w31"/>
</p>
```

Source: [210]

This use of <join> is TEI Conformant.

The major advantage of fragmentation and virtual joins is that it allows all the hierarchies in the text to be handled explicitly: both the privileged one directly represented and the alternate hierarchy that has been split up and rejoined. The major disadvantages are that (like most of the other methods described here) it privileges

one hierarchy over the others, requires special processing to reconstitute the elements of the other hierarchies, and, except in the case of <join>, can be semantically misleading.

20.4 Stand-off Markup

Most markup is characterized by the embedding of elements in the text. An alternative approach separates the text and the elements used to describe it. This approach is known as stand-off markup (see section *16.9. Stand-off Markup*). It establishes a new hierarchy by building a new tree whose nodes are XML elements that do not contain textual content, but rather links to another layer: *a node in another XML document or a span of text*. This approach can be subdivided according to different criteria. A first distinction concerns the link base, i.e. the content to which annotations are to be applied. Sometimes the link target contains markup that can be referred to explicitly, as in the following example where the offset markup uses the **xml:id** values on <w> to provide targets for <xi:include>[3]:

```
<l>
  <w  xml:id="w001">Scorn</w>
  <w  xml:id="w002">not</w>
  <w  xml:id="w003">the</w>
  <w  xml:id="w004">sonnet</w>; <w  xml:id="w005">critic</w>, <w  xml:id="w006">you</w>
  <w  xml:id="w007">have</w>
  <w  xml:id="w008">frowned</w>,
</l>
<l>
  <w  xml:id="w009">Mindless</w>
  <w  xml:id="w010">of</w>
  <w  xml:id="w011">its</w>
  <w  xml:id="w012">just</w>
  <w  xml:id="w013">honours</w>; <w  xml:id="w014">with</w>
  <w  xml:id="w015">this</w>
  <w  xml:id="w016">key</w>
</l>
<l>
  <w  xml:id="w017">Shakespeare</w>
  <w  xml:id="w018">unlocked</w>
  <w  xml:id="w019">his</w>
  <w  xml:id="w020">heart</w>; <w  xml:id="w021">the</w>
  <w  xml:id="w022">melody</w>
</l>
<l>
  <w  xml:id="w023">Of</w>
  <w  xml:id="w024">this</w>
  <w  xml:id="w025">small</w>
  <w  xml:id="w026">lute</w>
  <w  xml:id="w027">gave</w>
  <w  xml:id="w028">ease</w>
  <w  xml:id="w029">to</w>
  <w  xml:id="w030">Petrarch's</w>
  <w  xml:id="w031">wound</w>.
</l>
<!-- elsewhere in the document or in another document indicated by the value of @href -->
<p>
  <seg>
    <include xmlns="http://www.example.org/ns/nonTEI"
      href="."
```

[3] A fake namespace is given for XInclude here, to avoid the markup being interpreted literally during processing.

```
      xpointer="range(element(w001),element(w004))"/>
  </seg>
  <seg>
    <include xmlns="http://www.example.org/ns/nonTEI"
     href="."
      xpointer="range(element(w005),element(w013))"/>
  </seg>
  <seg>
    <include xmlns="http://www.example.org/ns/nonTEI"
     href="."
      xpointer="range(element(w014),element(w020))"/>
  </seg>
  <seg>
    <include xmlns="http://www.example.org/ns/nonTEI"
     href="."
      xpointer="range(element(w021),element(w031))"/>
  </seg>
</p>
```

Source: [210]

This is very similar to the use of <join> discussed above. The main advantages of the stand-off method are that it is possible to specify attributes on the aggregate <seg> elements, and that there exists off-the-shelf software that will perform appropriate processing. Stand-off markup may be used even when the base text being annotated is plain text, i.e. does not have any XML encoding. In this case, the range of text to be marked up is indicated by character offsets (see *16.2.4. TEI XPointer Schemes*, in particular *16.2.4.5. string-range(pointer, offset [, length])*). Another distinction concerns the number of files which can serve as link targets. Often, one (dedicated) annotation is used as the link target of all the other annotations. It is also possible to freely interlink several layers.

It has been noted that stand-off markup has several advantages over embedded annotations. In particular, it is possible to produce annotations of a text even when the source document is read-only. Furthermore, annotation files can be distributed without distributing the source text. Further advantages mentioned in the literature are that discontinuous segments of text can be combined in a single annotation, that independent parallel coders can produce independent annotations, and that different annotation files can contain different layers of information. Lastly, it has also been noted that this approach is elegant.

But there are also several drawbacks. First, new stand-off annotated layers require a separate interpretation, and the layers — although separate — depend on each other. Moreover, although all of the information of the multiple hierarchies is included, the information may be difficult to access using generic methods.

Inasmuch as it uses elements not included in the TEI namespace, stand-off markup involves an extension of the TEI.

20.5 Non-XML-based Approaches

There exist many non-XML methods of encoding a text that either solve or do not suffer the problem of the inability to encode overlapping hierarchies. These include, but are not limited to, the following proposals.

- Applying the notion of concurrent markup to XML (Hilbert et al. (2005)). This reintroduces the CONCUR feature of SGML, which was omitted from the XML specification.
- Designing a form of document representation in which several trees share all or part of the same frontier, and in which each individual view of the document has the form of a tree (see Dekhtyar and Iacob (2005)).

- The 'colored XML' proposal (Jagadish et al. (2004)), which stores a body of information as a set of intertwined XML trees. This approach eliminates unnecessary redundancy and makes the database readily updatable, while allowing the user to exploit different hierarchical access paths.
- The MultiX proposal (Chatti et al. (2007)) , which represents documents as directed graphs. Because XML is used to represent the graph, the document is, at least in principle, manipulable with standard XML tools.
- The Just-In-Time-Trees proposal (Durusau and O'Donnell (2002)), which stores documents using XML, but processes the XML representation in non-standard ways and allows it to be mapped onto data structures that are different from those known from XML.
- The LMNLLayered Markup and Annotation Language proposal. This offers alternatives to the basic XML linear form as well as its data and processing models. It uses an alternative notation to XML and a data structure based on Core Range Algebra (Tennison and Piez (2002)).
- Markup Languages for Complex DocumentsMLCD. This provides a notation (TexMECS) and a data structure (Goddag) as well as a draft constraint language for the representation of non-hierarchical structures; see Huitfeldt and Sperberg-McQueen (2001).

These approaches are based either on non-standard XML processing or data models, or not based on XML at all. Since TEI is currently based on XML they are not described any further in these Guidelines. Use of these methods with the TEI will certainly involve extensions; in most cases the documents will also be non-conformant.

Chapter 21

Certainty and Responsibility

Encoders of text often find it useful to indicate that some aspects of the encoded text are problematic or uncertain, and to indicate who is responsible for various aspects of the markup of the electronic text. These Guidelines provide three methods of recording uncertainty about the text or its markup:

- the <note> element defined in section *3.8. Notes, Annotation, and Indexing* may be used with a value of **certainty** for its **type** attribute.
- the <certainty> element defined in this chapter may be used to record the nature and degree of the uncertainty in a more structured way.
- the <alt> element defined in the module for linking and segmentation may be used to provide alternative encodings for parts of a text, as described in section *16.8. Alternation*.

There are three methods of indicating responsibility for different aspects of the electronic text:

- the TEI header records who is responsible for an electronic text by means of the <respStmt> element and other more specific elements (<author>, <sponsor>, <funder>, <principal>, etc.) used within the <titleStmt>, <editionStmt>, and <revisionDesc> elements.
- the <note> element may be used with a value of **resp** or **responsibility** in its **type** attribute.
- the <respons> element defined in this chapter may be used to record fine-grained structured information about responsibility for individual tags in the text.

No special steps are needed to use the <note> and <respStmt> elements, since they are defined in the core module and header respectively. The <alt> element is only available when the module for linking has been selected, as described in chapter *16. Linking, Segmentation, and Alignment*. To use the <certainty> and <respons> elements, the module for certainty and responsibility must be selected.

21.1 Levels of Certainty

Many types of uncertainty may be distinguished. The <certainty> element is designed to encode the following sorts:

- a given tag may or may not correctly apply (e.g. a given word may be a personal name, or perhaps not)
- the precise point at which an element begins or ends is uncertain
- the value to be given for an attribute is uncertain
- content supplied by the encoder (such as the expansion of an abbreviation marked by the <abbr> tag) is uncertain

- the transcription of a source text is uncertain, perhaps because it is hard to read or hard to hear; this sort of uncertainty is also handled by the <unclear> element in section *11.5.1. Damage, Illegibility, and Supplied Text*

The following types of uncertainty are *not* indicated with the <certainty> element:

- a number or date is imprecise
- the text is ambiguous, so a given passage has several possible interpretations
- a transcriber, editor, or author wishes to indicate a level of confidence in a factual assertion made in the text
- an author is not sure if the sentence she has chosen to start a paragraph is really the one she wants to retain in the final version

Precision of numbers and dates is discussed in section *3.5. Names, Numbers, Dates, Abbreviations, and Addresses*; well-defined ambiguity is handled with alternations in feature-structure values in chapter *18. Feature Structures*. Uncertainty about the truth of assertions in the text and other sorts of authorial and editorial uncertainty about whether the content is satisfactory are not handled by the <certainty> element, though they may be expressed using the <note> element.

21.1.1 Using Notes to Record Uncertainty

The simplest way of recording uncertainty about markup is to attach a note to the element or location about which one is unsure. In the following (invented) paragraph, for example, an encoder might be uncertain whether to mark 'Essex' as a place name or a personal name, since both might be plausible in the given context:

> Elizabeth went to Essex. She had always liked Essex.

Using <note>, the uncertainty here may be recorded quite simply:

```
<persName>Elizabeth</persName> went to <placeName>Essex</placeName>. She had always liked
<placeName>Essex</placeName>.
<note type="uncertainty" resp="#MSM">It is not
clear here whether <mentioned>Essex</mentioned>
refers to the place or to the nobleman. -MSM</note>
```

Using the normal mechanisms, the note may be associated unambiguously with specific elements of the text, thus:

```
<persName>Elizabeth</persName> went to <placeName xml:id="CE-p1a">Essex</placeName>.
She had always liked <placeName xml:id="CE-p1b">Essex</placeName>.
<note type="uncertainty" resp="#MSM" target="#CE-p1a #CE-p1b">It
is not clear here whether <mentioned>Essex</mentioned>
refers to the place or to the nobleman. If the latter,
it should be tagged as a personal name. -<name xml:id="MSM">Michael</name>
</note>
```

The advantage of this technique is its relative simplicity. Its disadvantage is that the nature and degree of uncertainty are not conveyed in any systematic way and thus are not susceptible to any sort of automatic processing.

21.1.2 Structured Indications of Uncertainty

To record uncertainty in a more structured way, susceptible of at least simple automatic processing, the <certainty> element may be used:

<certainty> indicates the degree of certainty or uncertainty associated with some aspect of the text markup.

Returning to the example, the <certainty> element may be used to record doubts about the proper encoding of 'Essex' in several ways of varying precision. To record merely that we are not certain that 'Essex' is in fact a place name, as it is tagged, we use the **target** attribute to identify the element in question, and the **locus** attribute to indicate what aspect of the markup we are uncertain about (in this case, whether we have used the correct 'gi', that is, element type, to mark it):

```
Elizabeth went to
<placeName xml:id="CE-pl1">Essex</placeName>.

<!-- ... elsewhere in the document ... -->
<certainty target="#CE-pl1" locus="gi">
  <desc>possibly not a placename</desc>
</certainty>
```

Because it is linked to the location of the uncertainty by a reference, the <certainty> element will typically be included in the same document as its target. It may be placed adjacent to the target element, or elsewhere in the document.

To record the further information that we estimate, subjectively, that there is a 60 percent chance of 'Essex' being a place name here, we can add a value for our **degree** of confidence (usually a number between 0 and 1, representing the estimated probability):

```
<!-- ... --><certainty target="#CE-pl1" locus="gi" degree="0.6"/>
```

According to one expert, there is a 60 percent chance of 'Essex' being a place name here, and a 40 percent chance of its being a personal name. We can use two <certainty> elements to indicate the two probabilities independently. Both elements indicate the same location in the text, but the second provides an alternative choice of generic identifier (in this case <persName>), which is given as the value of the **assertedValue** attribute:

```
<!-- ... --><certainty target="#CE-pl1" locus="gi" degree="0.6">
  <desc>probably a placename, but possibly not</desc>
</certainty>
<certainty
   target="#CE-pl1"
   locus="gi"
   degree="0.4"
   assertedValue="persName">
  <desc>may refer to the Earl of Essex</desc>
</certainty>
```

Finally, we may wish to make our probability estimates contingent on some condition. In the passage 'Elizabeth went to Essex; she had always liked Essex,' for example, we may feel there is a 60 percent chance that the county is meant, and a 40 percent chance that the earl is meant. But the two occurrences of the word are not independent: there is (we may feel) no chance at all that one occurrence refers to the county and one to the earl. We can express this by using the **given** attribute to list the identifiers of <certainty> elements.

```
Elizabeth went to <placeName  xml:id="CE-PL1">Essex</placeName>.
She had always liked <placeName  xml:id="CE-PL2">Essex</placeName>.

<!-- ...  -->
<!-- 60% chance that P1 is a placename, 40% chance a personal name.  -->
<certainty
   xml:id="cert-1"
   target="#CE-PL1"
   locus="gi"
   degree="0.6">
  <desc>probably a placename, but possibly not"</desc>
</certainty>
<certainty
   xml:id="cert-2"
   target="#CE-PL1"
   locus="gi"
   assertedValue="persName"
   degree="0.4">
  <desc>may refer to the Earl of Essex"</desc>
</certainty>
<!-- 60% chance that P2 is a placename, 40% chance a personal name.  100% chance that it agrees with P1.  -->
<certainty
   target="#CE-PL2"
   locus="gi"
   given="#cert-1"
   degree="1.0">
  <desc>if P1 is a placename, P2 certainly is"</desc>
</certainty>
<certainty
   target="#CE-PL2"
   locus="gi"
   assertedValue="persName"
   degree="1.0"
   given="#cert-2">
  <desc>if p1 refers to the Earl of Essex, so does P2</desc>
</certainty>
```

When **given** conditions are listed, the <certainty> element is interpreted as claiming a given degree of confidence in a particular markup given the assertional content of the <certainty> elements indicated—that is, *if the markup described in the indicated <certainty> elements is correct.*

Conditional confidence may be less that 100 percent: given the sentence 'Ernest went to old Saybrook', we may interpret 'Saybrook' as a personal name or a place name, assigning a 60 percent probability to the former. If it is a place name, there may be a 50 percent chance that the place name actually in question is 'Old Saybrook' rather than 'Saybrook', while if it is correctly tagged as a personal name, it is much more likely (say, 90 percent certain) that the name is 'Saybrook'. Hence there is uncertainty about the correct location for the markup as well as about which markup to use. This state of affairs can be expressed using the <certainty> element thus:

```
Ernest went to <anchor  xml:id="CE-a1"/> old <persName  xml:id="CE-p2">Saybrook</persName>.

<certainty
   xml:id="cert1"
   target="#CE-p2"
   locus="gi"
   degree="0.6"/>
```

```
<certainty
   target="#CE-p2"
   locus="startLoc"
   given="#cert1"
   degree="0.9"/>
<certainty
   xml:id="cert2"
   target="#CE-p2"
   locus="gi"
   assertedValue="placeName"
   degree="0.4"/>
<certainty
   target="#CE-p2"
   locus="startLoc"
   given="#cert2"
   degree="0.5"/>
<certainty
   xml:id="cert3"
   target="#CE-p2"
   locus="startLoc"
   assertedValue="CE-a1"
   given="#cert1"
   degree="0.1"/>
<certainty
   xml:id="cert4"
   target="#CE-p2"
   locus="startLoc"
   assertedValue="CE-a1"
   given="#cert2"
   degree="0.5"/>
```

Note the use of the **assertedValue** on <certainty> elements **cert3** and **cert4** to reference the <anchor> element placed at the alternative starting point for the element.

Multiplying the numeric values out, this markup may be interpreted as assigning specific probabilities to three different ways of marking up the sentence:

```
Ernest went to old <persName>Saybrook</persName>. (0.6 * 0.9, or 0.54)
Ernest went to old <placeName>Saybrook</placeName>. (0.4 * 0.5, or 0.20)
Ernest went to <placeName>old Saybrook</placeName>. (0.4 * 0.5, or 0.20)
```

The probabilities do not add up to 1.00 because the markup indicates that if 'Saybrook' is (part of) a personal name, there is a 10 percent likelihood that the element should start somewhere other than the place indicated, without however giving an alternative location; there is thus a 6 percent chance (0.1×0.6) that none of the alternatives given is correct.

If an attribute value is uncertain, the **locus** attribute takes as its value the name of the attribute in question. In this example, there is only a 50 percent chance that the question was spoken by participant A:

```
<u xml:id="CE-u1" who="#A">Have you heard the election results?</u>
<certainty target="#CE-u1" locus="att.who" degree="0.5"/>
```

Doubts about whether the transcription is correct may be expressed by assigning to **locus** the value *transcribedContent*. For example, if the source is hard to read and so the transcription is uncertain:

```
I have a <emph  xml:id="CE-p3">gub</emph>.

<certainty  target="#CE-p3"  locus="transcribedContent"  degree="0.5"/>
```

Degrees of confidence in the proper expansion of abbreviations may also be expressed, by using the value *suppliedContent*:

```
You will want to use
<choice>
  <expan  xml:id="CE-e1">Standard
     Generalized Markup Language</expan>
  <expan  xml:id="CE-e4">Some Grandiose Methodology for Losers</expan>
  <abbr>SGML</abbr>
</choice> ...

<!-- ... -->
<certainty  target="#CE-e1"  locus="suppliedContent"  degree="0.9"/>
```

The **assertedValue** attribute should be used to provide an alternative value for whatever aspect of the markup is in doubt: an alternative generic identifier, or the identifier of an alternative starting or ending point, as already shown, an alternative attribute value, or alternative element content, as in this example:

```
I have a <emph  xml:id="CE-P3">gub</emph>.

<certainty
   target="#CE-P3"
   locus="transcribedContent"
   assertedValue="gun"
   degree="0.8">
  <desc>a gun makes more sense in a holdup</desc>
</certainty>
```

Since attribute values have no internal substructure, the **assertedValue** attribute is useful for specifying alternative transcriptions only in relatively restricted circumstances (specifically, when the alternative reading has no elements nested within it). More robust methods of handling uncertainties of transcription are the <unclear> element and the <app> and <rdg> elements described in chapter *12. Critical Apparatus*. The <certainty> element allows for indications of uncertainty to be structured with at least as much detail and clarity as appears to be currently required in most ongoing text projects. It is expected that in the future more adequate systems for expressing uncertainty will be developed. These may extend the <certainty> element or they may make use of the feature-structure encoding mechanisms described in chapter *18. Feature Structures*.

The <certainty> element and the other TEI mechanisms for indicating uncertainty provide a range of methods of graduated complexity. Simple expressions of uncertainty may be made by using the <note> element. This is simple and convenient, and can accommodate either a discursive and unstructured indication of uncertainty, or a complex and structured but probably project-specific expression of uncertainty. In general, however, unless special steps are taken, the <note> element does not provide as much expressive power as the <certainty> element, and in cases where highly structured certainty information must be given, it is recommended that the <certainty> element be used.

The <certainty> element may be used for simple unqualified indications of uncertainty, in which case only the **locus** and **target** attributes might be specified. In more complex cases, the other attributes may be used to provide fuller information. While these attributes may take any string of characters as value, the recommended

values should be used wherever possible; if they are not appropriate in a given situation, encoders should provide their own controlled vocabulary and document it in the <encodingDesc> or <tagUsage> elements of the TEI header.

21.2 Attribution of Responsibility

In general, attribution of responsibility for the transcription and markup of an electronic text is made by <respStmt> elements within the header: specifically, within the title statement, the edition statement(s), and the revision history.

In some cases, however, more detailed element-by-element information may be desired. For example, an encoder may wish to distinguish between the individuals responsible for transcribing the content and those responsible for determining that a given word or phrase constitutes a proper noun. Where such fine-grained attribution of responsibility is required, the <respons> element can be used:

<respons> (responsibility) identifies the individual(s) responsible for some aspect of the markup of particular element(s).

This element allows one or more aspects of the markup to be attributed to a given individual. The **target** and **locus** attributes function as they do on the <certainty> element described in section *21.1. Levels of Certainty*: the **target** attribute points at a particular element (or set of elements), and **locus** indicates the particular aspect of the encoding of those elements for which responsibility is to be assigned. The suggested values may be combined as appropriate. For example, to indicate that RC is responsible for transcribing an illegible word, and that PMWR is responsible for identifying that word as a proper noun, the text might be encoded thus:

```
Earnest went to old <persName xml:id="CE-p5">Saybrook</persName>.

<!-- ... -->
<respons target="#CE-p5" locus="transcribedContent" resp="#RC"/>
<respons target="#CE-p5" locus="gi location" resp="#PMWR"/>
<list type="encoders">
  <item xml:id="PMWR"/>
  <item xml:id="RC"/>
</list>
```

Some elements bear specialized **resp** or **agent** attributes, which have specific meanings that vary from element to element; the <respons> element should be reserved for the general aspects of responsibility common to all text transcription and markup, and should not be confused with the more specific attributes on individual elements.

21.3 The Certainty Module

The module described in this chapter makes available the following additional elements:

Module certainty: Certainty and uncertainty

- Elements defined: certainty respons

The selection and combination of modules to form a TEI schema is described in *1.2. Defining a TEI Schema*.

Chapter 22

Documentation Elements

This chapter describes a module which may be used for the documentation of the XML elements and element classes which make up any markup scheme, in particular that described by the TEI Guidelines, and also for the automatic generation of schemas or DTDs conforming to that documentation. It should be used also by those wishing to customize or modify these Guidelines in a conformant manner, as further described in chapters *23.2. Personalization and Customization* and *23.3. Conformance* and may also be useful in the documentation of any other comparable encoding scheme, even though it contains some aspects which are specific to the TEI and may not be generally applicable.

An overview of the kind of processing environment envisaged for the module described by this chapter may be helpful. In the remainder of this chapter we refer to software which provides such a processing environment as an ODD processor.[1] Like any other piece of XML software, an ODD processor may be instantiated in many ways: the current system uses a number of XSLT stylesheets which are freely available from the TEI, but this specification makes no particular assumptions about the tools which will be used to provide an ODD processing environment.

As the name suggests, an ODD processor uses a single XML document to generate multiple outputs. These outputs will include:

- formal reference documentation for elements, attributes, element classes, patterns, etc. such as those provided in *Appendix C Elements* below;
- detailed descriptive documentation, embedding some parts of the formal reference documentation, such as the tag description lists provided in this and other chapters of these Guidelines;
- declarative code for one or more XML schema languages, specifically RELAX NG or W3C Schema.
- declarative code for fragments which can be assembled to make up an XML Document Type Declaration.

The input required to generate these outputs consists of running prose, and special purpose elements documenting the components (elements, classes, etc.) which are to be declared in the chosen schema language. All of this input is encoded in XML using the module defined by this chapter. In order to support more than one schema language, this module uses a comparatively high-level model which can then be mapped by an ODD processor to the specific constructs appropriate to the schema language in use. Although some modern schema languages such as RELAX NG or W3C Schema natively support self-documentary features of this kind, we have chosen to retain the ODD model, if only for reasons of compatibility with earlier versions of these Guidelines. We do however use the ISO standard XML schema language RELAX NG (`http://www.relaxng.org`) as a means of declaring content models, rather than inventing a completely new XML-based representation for them.

[1] ODD is short for 'One Document Does it all', and was the name invented by the original TEI Editors for the predecessor of the system currently used for this purpose. See further Burnard and Sperberg-McQueen (1995) and Burnard and Rahtz (2004).

In the TEI abstract model, a markup scheme (a schema) consists of a number of discrete modules, which can be combined more or less as required. Each major chapter of these Guidelines defines a distinct module. Each module declares a number of elements specific to that module, and may also populate particular classes. All classes are declared globally; particular modules extend the meaning of a class by adding elements or attributes to it. Wherever possible, element content models are defined in terms of classes rather than in terms of specific elements. Modules can also declare particular patterns, which act as short-cuts for commonly used content models or class references.

In the present chapter, we discuss the elements needed to support this system. In addition, section *22.1. Phrase Level Documentary Elements* discusses some general purpose elements which may be useful in any kind of technical documentation, wherever there is need to talk about technical features of an XML encoding such as element names and attributes. Section *22.2. Modules and Schemas* discusses the elements which are used to document XML modules and their high-level components. Section *22.3. Specification Elements* discusses the elements which document XML elements and their attributes, element classes, and generic patterns or macros. Finally, section *22.7. Module for Documention Elements* gives an overview of the whole module.

22.1 Phrase Level Documentary Elements

22.1.1 Phrase Level Terms

In any kind of technical documentation, the following phrase-level elements may be found useful for marking up strings of text which need to be distinguished from the running text because they come from some formal language:

<code> contains literal code from some formal language such as a programming language.
  @**lang** (formal language) a name identifying the formal language in which the code is expressed

<ident> (identifier) contains an identifier or name for an object of some kind in a formal language.

Like other phrase-level elements used to indicate the semantics of a typographically distinct string, these are members of the model.emph class. They are available anywhere that running prose is permitted when the module defined by this chapter is included in a schema.

The <code> and <ident> elements are intended for use when citing brief passages in some formal language such as a programming language, as in the following example:

```
<p>If the variable <ident>z</ident> has a value of zero, a statement
such as <code>x=y/z</code> will usually cause a fatal error.</p>
```

If the cited phrase is a mathematical or chemical formula, the more specific <formula> element defined by the figures module (*14.2. Formulæ and Mathematical Expressions*) may be more appropriate.

A further group of similar phrase-level elements is also defined for the special case of representing parts of an XML document:

<att> (attribute) contains the name of an attribute appearing within running text.

<gi> (element name) contains the name (generic identifier) of an element.

<tag> contains text of a complete start- or end-tag, possibly including attribute specifications, but excluding the opening and closing markup delimiter characters.

<val> (value) contains a single attribute value.

These elements constitute the model.phrase.xml class, which is also a subclass of model.phrase. They are also available anywhere that running prose is permitted when the module defined by this chapter is included in a schema.

As an example of the recommended use of these elements, we quote from an imaginary TEI working paper:

```
<p>The <gi>gi</gi> element is used to tag
element names when they appear in the text; the
<gi>tag</gi> element however is used to show how a tag as
such might appear. So one might talk of an occurrence of the
<gi>blort</gi> element which had been tagged
<tag>blort type='runcible'</tag>. The
<att>type</att> attribute may take any name token as
value; the default value is <val>spqr</val>, in memory of
its creator.</p>
```

Within technical documentation, it is also often necessary to provide more extended examples of usage or to present passages of markup for discussion. The following special elements are provided for these purposes:

<eg> (example) contains any kind of illustrative example.

<egXML> (example of XML) contains a single well-formed XML fragment demonstrating the use of some XML element or attribute, in which the <egXML> element itself functions as the root element.

Like the <code> element, the <egXML> element is used to mark strings of formal code, or passages of XML markup. The <eg> element may be used to enclose any kind of example, which will typically be rendered as a distinct block, possibly using particular formatting conventions, when the document is processed. It is a specialised form of the more general <q> element provided by the TEI core module. In documents containing examples of XML markup, the <egXML> element should be used for preference, as further discussed below in *22.4.2. Exemplification of Components*, since the content of this element can be checked for well-formedness.

These elements are members of the class att.xmlspace which provides the following attribute:

att.xmlspace groups TEI elements for which it is reasonable to specify whitespace management using the W3C-defined xml:space attribute.

@xml:space signals an intention that white space should be preserved by applications

These elements are added to the class model.egLike when this module is included in a schema. That class is a part of the general model.inter class, thus permitting <eg> or <egXML> elements to appear either within or between paragraph-like elements.

22.1.2 Element and Attribute Descriptions

Within the body of a document using this module, the following elements may be used to reference parts of the specification elements discussed in section *22.3. Specification Elements*, in particular the brief prose descriptions these provide for elements and attributes.

<specList> (specification list) marks where a list of descriptions is to be inserted into the prose documentation.

<specDesc/> (specification description) indicates that a description of the specified element or class should be included at this point within a document.

TEI practice requires that a <specList> listing the elements under discussion introduce each subsection of a module's documentation. The source for the present section, for example, begins as follows:

```
<div3>
  <head>Element and attribute descriptions</head>
  <p>Within the body of a document using this module, the following
     elements may be used to reference parts of the specification elements
     discussed in section <ptr  target="#TDcrystals"/>, in particular the
     brief prose descriptions these provide for elements and attributes.
```

```
  <specList>
     <specDesc  key="specList"/>
     <specDesc  key="specDesc"/>
   </specList>
  </p>
  <p>TEI practice requires that a <gi>specList</gi> listing the elements
     ...
  </p>
<!-- ... -->
</div3>
```

When formatting the <ptr> element in this example, an ODD processor might simply generate the section number and title of the section referred to, perhaps additionally inserting a link to the section. In a similar way, when processing the <specDesc> elements, an ODD processor must recover relevant details of the elements being specified (<specList> and <specDesc> in this case) from their associated declaration elements: typically, the details recovered will include a brief description of the element and its attributes. These, and other data, will be stored in a specification element elsewhere within the current document, or they may be supplied by the ODD processor in some other way, for example from a database. For this reason, the link to the required specification element is always made using a TEI-defined key rather than an XML IDREF value. The ODD processor uses this key as a means of accessing the specification element required. There is no requirement that this be performed using the XML ID/IDREF mechanism, but there is an assumption that the identifier be unique.

A <specDesc> generates in the documentation the identifier, and also the contents of the <desc> child of whatever specification element is indicated by its **key** attribute, as in the example above. Documentation for any attributes specified by the **atts** attribute will also be generated as an associated attribute list, .

22.2 Modules and Schemas

As mentioned above, the primary purpose of this module is to facilitate the documentation of an XML schema derived from the TEI Guidelines. The following elements are provided for this purpose:

<schemaSpec> (schema specification) generates a TEI-conformant schema and documentation for it.

<moduleSpec> (module specification) documents the structure, content, and purpose of a single module, i.e. a named and externally visible group of declarations.

<moduleRef> (module reference) references a module which is to be incorporated into a schema.

<specGrp> (specification group) contains any convenient grouping of specifications for use within the current module.

<specGrpRef/> (reference to a specification group) indicates that the declarations contained by the <specGrp> referenced should be inserted at this point.

<attRef/> (attribute pointer) points to the definition of an attribute or group of attributes.

A module is a convenient way of grouping together element and other declarations and associating an externally-visible name with the group. A specification group performs essentially the same function, but the resulting group is not accessible outside the scope of the ODD document in which it is defined, whereas a module can be accessed by name from any TEI schema.Modules, elements, and their attributes, element classes, and patterns are all individually documented using further elements described in section *22.3. Specification Elements* below; part of that specification includes the name of a module to which the component belongs. An ODD processor generating XML DTD or schema fragments from a document marked up according to the recommendations of this chapter will generate such fragments for each <moduleSpec> element found. For example, the chapter documenting the TEI module for names and dates contains a module specification like the following:

```
<moduleSpec xml:id="XDND" ident="namesdates">
  <altIdent type="FPI">Names and Dates</altIdent>
  <desc>Additional elements for names and dates</desc>
</moduleSpec>
```

together with specifications for all the elements, classes, and patterns which make up that module, expressed using <elementSpec>, <classSpec>, or <macroSpec> elements as appropriate. (These elements are discussed in section *22.3. Specification Elements* below.) Each of those specifications carries a **module** attribute, the value of which is `namesdates`. An ODD processor encountering the <moduleSpec> element above can thus generate a schema fragment for the TEI namesdates module that includes declarations for all the elements (etc.) which reference it.

In most realistic applications, it will be desirable to combine more than one module together to form a complete schema. A schema consists of references to one or more modules or specification groups, and may also contain explicit declarations or redeclarations of elements (see further *22.5. Building a Schema*). Any combination of modules can be used to create a schema: the distinction between base and additional tagsets in earlier versions of the TEI scheme has not been carried forward into P5.

A schema can combine references to TEI modules with references to other (non-TEI) modules using different namespaces, for example to include mathematical markup expressed using MathML in a TEI document. By default, the effect of combining modules is to allow all of the components declared by the constituent modules to coexist (where this is syntactically possible: where it is not — for example, because of name clashes — a schema cannot be generated). It is also possible to over-ride declarations contained by a module, as further discussed in section *22.5. Building a Schema*

It is often convenient to describe and operate on sets of declarations smaller than the whole, and to document them in a specific order: such collections are called specGrps (specification groups). Individual <specGrp> elements are identified using the global **xml:id** attribute, and may then be referenced from any point in an ODD document using the <specGrpRef> element. This is useful if, for example, it is desired to describe particular groups of elements in a specific sequence. Note however that the order in which element declarations appear within the schema code generated from a <moduleSpec> element is not in general affected by the order of declarations within a <specGrp>.

An ODD processor will generate a piece of schema code corresponding with the declarations contained by a <specGrp> element in the documentation being output, and a cross-reference to such a piece of schema code when processing a <specGrpRef>. For example, if the input text reads

```
<p>This module contains three red elements:
<specGrp xml:id="RED">
   <elementSpec ident="beetroot">
<!-- ... -->
   </elementSpec>
   <elementSpec ident="east">
<!-- ... -->
   </elementSpec>
   <elementSpec ident="rose">
<!-- ... -->
   </elementSpec>
  </specGrp>
and two blue ones:
<specGrp xml:id="BLUE">
   <elementSpec ident="sky">
<!-- ... -->
   </elementSpec>
```

```
   <elementSpec ident="bayou">
<!-- ... -->
   </elementSpec>
  </specGrp>
</p>
```

then the output documentation will replace the two <specGrp> elements above with a representation of the schema code declaring the elements <beetroot>, <east>, and <rose> and that declaring the elements <sky> and <bayou> respectively. Similarly, if the input text contains elsewhere a passage such as

```
<div>
  <head>An overview of the imaginary module</head>
  <p>The imaginary module contains declarations for coloured things:
  <specGrpRef target="#RED"/>
    <specGrpRef target="#BLUE"/>
  </p>
</div>
```

then the <specGrpRef> elements may be replaced by an appropriate piece of reference text such as 'The RED elements were declared in section 4.2 above', or even by a copy of the relevant declarations. As stated above, the order of declarations within the imaginary module described above will not be affected in any way. Indeed, it is possible that the imaginary module will contain declarations not present in any specification group, or that the specification groups will refer to elements that come from different modules. Specification groups are always local to the document in which they are defined, and cannot be referenced externally (unlike modules).

22.3 Specification Elements

The following elements are used to specify elements, classes, and patterns for inclusion in a given module:

<elementSpec> (element specification) documents the structure, content, and purpose of a single element type.

<classSpec> (class specification) contains reference information for a TEI element class; that is a group of elements which appear together in content models, or which share some common attribute, or both.

@**generate** indicates which alternation and sequence instantiations of a model class may be referenced. By default, all variations are permitted.

<macroSpec> (macro specification) documents the function and implementation of a pattern.

Unlike most elements in the TEI scheme, each of these elements has a fairly rigid internal structure consisting of a large number of child elements which are always presented in the same order. For this reason, we refer to them metaphorically as 'crystals'. Furthermore, since these elements all describe markup objects in broadly similar ways, they have several child elements in common. In the remainder of this chapter, we discuss first the elements which are common to all the specification elements, and then those which are specific to a particular type.

Specification elements may appear at any point in an ODD document, both between and within paragraphs as well as inside a <specGrp> element, but the specification element for any particular component may only appear once (except in the case where a modification is being defined; see further *22.5. Building a Schema*). The order in which they appear will not affect the order in which they are presented within any schema module generated from the document. In documentation mode, however, an ODD processor will output the schema declarations corresponding with a specification element at the point in the text where they are encountered,

provided that they are contained by a <specGrp> element, as discussed in the previous section. An ODD processor will also associate all declarations found with the nominated module, thus including them within the schema code generated for that module, and it will also generate a full reference description for the object concerned in a catalogue of markup objects. These latter two actions always occur irrespective of whether or not the declaration is included in a <specGrp>.

22.4 Common Elements

This section discusses the child elements common to all of the specification elements. These child elements are used to specify the naming, description, exemplification, and classification of the specification elements.

22.4.1 Description of Components

<remarks> contains any commentary or discussion about the usage of an element, attribute, class, or entity not otherwise documented within the containing element.

<listRef> (list of references) supplies a list of significant references to places where this element is discussed, in the current document or elsewhere.

One or more <desc> elements defined by the core module may be used to provide a brief characterization of the intended function of the element, class, value etc. being documented, as in the following example:

```
<elementSpec module="drama" usage="opt" ident="actor">
  <desc>Name of an actor appearing within a cast list.</desc>
  <desc xml:lang="ja"> □□人□□□□□□□□□□□□□□□□</desc>
  <desc xml:lang="it">nome di un attore che appare nella lista dei personaggi.</desc>
<!-- ... -->
</elementSpec>
```

The <remarks> element contains any additional commentary about how the item concerned may be used, details of implementation-related issues, suggestions for other ways of treating related information etc., as in the following example:

```
<elementSpec module="core" ident="foreign">
<!--... -->
  <remarks>
    <p>This element is intended for use only where no other element
        is available to mark the phrase or words concerned. The global
    <att>xml:lang</att> attribute should be used in preference to this element
        where it is intended to mark the language of the whole of some text
        element.</p>
    <p>The <gi>distinct</gi> element may be used to identify phrases
        belonging to sublanguages or registers not generally regarded as true
        languages.</p>
  </remarks>
<!--... -->
</elementSpec>
```

A specification element will usually conclude with a list of references, each tagged using the standard <ptr> element, and grouped together into a <listRef> element: in the case of the <foreign> element discussed above, the list is as follows:

```
<listRef>
  <ptr target="#COHQHF"/>
</listRef>
```

where the value **COHQF** is the identifier of the section in the Guidelines where this element is fully documented.

22.4.2 Exemplification of Components

<exemplum> groups an example demonstrating the use of an element along with optional paragraphs of commentary.

<eg> (example) contains any kind of illustrative example.

<egXML> (example of XML) contains a single well-formed XML fragment demonstrating the use of some XML element or attribute, in which the <egXML> element itself functions as the root element.

The <exemplum> element is used to combine a single illustrative example with an optional paragraph of commentary following or preceding it. The illustrative example itself may be marked up using either the <eg> or the <egXML> element.

If an example contains XML markup, it should be marked up using the <egXML> element. In such a case, it will clearly be necessary to distinguish the markup within the example from the markup of the document itself. In an XML schema environment, this is easily done by using a different name space for the <egXML> element. For example:

```
<p>The <gi>term</gi> element may be used
to mark any technical term, thus :
<egXML xmlns="http://www.tei-c.org/ns/Examples">
This <term>recursion</term> is
giving me a headache.</egXML></p>
```

Alternatively, the XML tagging within an example may be 'escaped', either by using entity references, or by wrapping the whole example in a CDATA marked section:

```
<p>The <gi>term</gi> element may be used
to mark any technical term, thus :
<egXML xmlns="http://www.tei-c.org/ns/Examples">
This &lt;term&gt;recursion&lt;/term&gt; is
giving me a headache.</egXML></p>
```

or, equivalently:

```
<p>The <gi>term</gi> element may be used
to mark any technical term, thus :
<egXML xmlns="http://www.tei-c.org/ns/Examples"><![CDATA[
This <term>recursion</term> is
giving me a headache.]]></egXML></p>
```

However, escaping the markup in this way will make it impossible to validate, and should therefore generally be avoided.

If the XML contained in an example is not well-formed then it must either be enclosed in a CDATA marked section, or 'escaped' as above: this applies whether the <eg> or <egXML> is used. If it is well-formed but not valid, then it should be enclosed in a CDATA marked section within an <egXML>.

An <egXML> element should not be used to tag non-XML examples: the general purpose <eg> or <q> elements should be used for such purposes.

22.4.3 Classification of Components

In the TEI scheme elements are assigned to one or more classes, which may themselves have subclasses. The following elements are used to indicate class membership:

<classes> specifies all the classes of which the documented element or class is a member or subclass.

<memberOf> specifies class membership of the parent element or class.

@**key** specifies the identifier for a class of which the documented element or class is a member or subclass

The <classes> element appears within either the <elementSpec> or <classSpec> element. It specifies the classes of which the element or class concerned is a member by means of one or more <memberOf> child elements. Each such element references a class by means of its **key** attribute. Classes themselves are defined by the <classSpec> element described in section *22.4.6. Element Classes* below.

For example, to show that the element <gi> is a member of the class model.phrase.xml, the <elementSpec> which documents this element contains the following <classes> element:

```
<classes>
  <memberOf  key="model.phrase.xml"/>
</classes>
```

22.4.4 Element Specifications

The <elementSpec> element is used to document an element type, together with its associated attributes. In addition to the elements listed above, it may contain the following subcomponents:

<content> (content model) contains the text of a declaration for the schema documented.

<attList> contains documentation for all the attributes associated with this element, as a series of <attDef> elements.

@**org** (organization) specifies whether all the attributes in the list are available (org="group") or only one of them (org="choice")

The content of the element <content> may be expressed in one of two ways. It may use a schema language of some kind, as defined by a pattern called `macro.schemaPattern`, which is provided by the module defined in this chapter. Alternatively, the legal content for an element may be fully specified using the <valList> element, described in *22.4.5. Attribute List Specification* below.

In the case of the TEI Guidelines, element content models are defined using RELAX NG patterns, but the user may over-ride this by redefining this pattern.

Here is a very simple example

```
<content>
  <rng:text/>
</content>
```

This content model uses the RELAX NG namespace, and will be copied unchanged to the output when RELAX NG schemas are being generated. When an XML DTD is being generated, an equivalent declaration (in this case `(#PCDATA)`) will be output.

Here is a more complex example:

```
<content>
  <rng:group>
    <rng:ref  name="fileDesc"/>
    <rng:zeroOrMore>
      <rng:ref  name="model.headerPart"/>
    </rng:zeroOrMore>
    <rng:optional>
      <rng:ref  name="revisionDesc"/>
    </rng:optional>
  </rng:group>
</content>
```

This is the content model for the <teiHeader> element, expressed in the RELAX NG syntax, which again is copied unchanged to the output during schema generation. The equivalent DTD notation generated from this is `(fileDesc, (%model.headerPart;)*, revisionDesc?)`.

The RELAX NG language does not formally distinguish element names, attribute names, class names, or macro names: all names are patterns which are handled in the same way, as the above example shows. Within the TEI scheme, however, different naming conventions are used to distinguish amongst the objects being named. Unqualified names (`fileDesc`, `revisionDesc`) are always element names. Names prefixed with `model.` or `att.` (e.g. model.headerPart are always class names. In DTD language, classes are represented by parameter entities (`%model.headerPart;` in the above example); see further *1. The TEI Infrastructure*.

22.4.5 Attribute List Specification

The <attList> element is used to document information about a collection of attributes, either within an <elementSpec>, or within a <classSpec>. An attribute list can be organized either as a group of attribute definitions, all of which are understood to be available, or as a choice of attribute definitions, of which only one is understood to be available. An attribute list may also contain nested attribute lists.

The <attDef> element is used to document a single attribute, using an appropriate selection from the common elements already mentioned and the following which are specific to attributes:

<attDef> (attribute definition) contains the definition of a single attribute.
  @**usage** specifies the optionality of an attribute or element.

<datatype> specifies the declared value for an attribute, by referring to any datatype defined by the chosen schema language.

<defaultVal> (default value) specifies the default declared value for an attribute.

<valDesc> (value description) specifies any semantic or syntactic constraint on the value that an attribute may take, additional to the information carried by the datatype element.

<valList> (value list) contains one or more <valItem> elements defining possible values for an attribute.

<valItem> documents a single attribute-value within a list of possible or mandatory items.

The <attList> within an <elementSpec> is used to specify only the attributes which are specific to that particular element. Instances of the element may carry other attributes which are declared by the classes of which the element is a member. These extra attributes, which are shared by other elements, or by all elements, are specified by an <attList> contained within a <classSpec> element, as described in section *22.4.6. Element Classes* below.

22.4.5.1 Datatypes

The <datatype> element is used to state what kind of value an attribute may have, using whatever facilities are provided by the underlying schema language. For the TEI scheme, expressed in RELAX NG, elements from the RELAX NG namespace may be used, for example

```
<datatype>
  <rng:text/>
</datatype>
```

permits any string of Unicode characters not containing markup, and is thus the equivalent of `CDATA` in DTD language.

The RELAX NG language also provides support for a number of primitive datatypes which may be specified here, using the <rng:data> element: thus one may write

```
<datatype>
  <rng:data  type="Boolean"/>
</datatype>
```

to specify that an element or attribute's contents should conform to the W3C definition for Boolean.

Although only one child element may be given, this might be a selector such as **rng:choice** to indicate multiple possibilities:

```
<datatype>
  <rng:choice>
    <rng:data  type="Date"/>
    <rng:data  type="Float"/>
  </rng:choice>
</datatype>
```

which would permit either a date or a real number. In fact, the child element might be a `rng:list` element to indicate that a sequence of values is required, a `rng:param` element to specify a regular expression, or even a list of explicit `rng:value`s. Such usages are permitted by the scheme documented here, but are not recommended when it is desired to remain independent of a particular schema language, since the full generality of one schema language cannot readily be converted to that of another. In the TEI abstract model, datatyping should preferably be carried out either by explicit enumeration of permitted values (using the TEI-specific <valList> element described below), or by definition of an explicit pattern, using the TEI-specific <macroSpec> element discussed further in section *22.4.7. Pattern Documentation*.

22.4.5.2 Value Specification

The <valDesc> element may be used to describe constraints on data content in an informal way: for example

```
<valDesc>must point to another <gi>align</gi>
element logically preceding this
one.</valDesc>
```

```
<valDesc>Values should be Library of Congress subject
headings.</valDesc>
```

```
<valDesc>A bookseller's surname,
taken from the list in <title>Pollard and Redgrave</title>
</valDesc>
```

As noted above, the <datatype> element constrains the possible values for an attribute. The <valDesc> element can be used to describe further constraints. For example, to specify that an attribute **age** can take positive integer values less than 100, the datatype data.numeric might be used in combination with a <valDesc> such as 'values must be positive integers less than 100'.

More usually, however, where constraints on values are explicitly enumerated, the <valList> element is used, as in the following example:

```
<valList type="closed">
  <valItem ident="req">
    <gloss>required</gloss>
  </valItem>
  <valItem ident="mwa">
    <gloss>mandatory when applicable</gloss>
  </valItem>
  <valItem ident="rec">
    <gloss>recommended</gloss>
  </valItem>
  <valItem ident="rwa">
    <gloss>recommended when applicable</gloss>
  </valItem>
  <valItem ident="opt">
    <gloss>optional</gloss>
  </valItem>
</valList>
```

Since this value list specifies that it is of type **closed**, only the values enumerated and glossed above are legal, and an ODD processor will typically enforce these constraints in the schema fragment generated.

The <valList> element is also used to provide illustrative examples of the kinds of values expected. In such cases the **type** attribute will have the value **open** and the datatype will usually be data.enumerated.

Note that the <gloss> element is needed to explain the significance of the identifier for an item only when this is not apparent, for example because it is abbreviated, as in the above example. It should not be used to provide a full description of the intended meaning (this is the function of the <desc> element), nor to comment on equivalent values in other schemes (this is the purpose of the <equiv> element).

22.4.5.3 Examples

The following <attList> demonstrates some of the possibilities; for more detailed examples, consult the tagged version of the reference material in these Guidelines.

```
<attList>
  <attDef ident="type">
    <desc>describes the form of the list.</desc>
    <datatype>
      <rng:text/>
    </datatype>
    <defaultVal>simple</defaultVal>
    <valList type="semi">
      <valItem ident="ordered">
        <desc>list items are numbered or lettered. </desc>
      </valItem>
      <valItem ident="bulleted">
        <desc>list items are marked with a bullet or other
                typographic device. </desc>
      </valItem>
```

```
    <valItem  ident="simple">
      <desc>list items are not numbered or bulleted.</desc>
    </valItem>
    <valItem  ident="gloss">
      <desc>each list item glosses some term or
             concept, which is given by a label element preceding
             the list item.</desc>
    </valItem>
   </valList>
   <remarks>
    <p>The formal syntax of the element declarations allows
    <gi>label</gi> tags to be omitted from lists tagged <tag>list
             type="gloss"</tag>; this is however a semantic error.</p>
   </remarks>
  </attDef>
</attList>
```

In the following example, the **org** attribute is used to indicate that instances of the element concerned may bear either a **bar** attribute or a **baz** attribute, but not both. The **bax** attribute is always available:

```
<attList>
  <attDef  ident="bax">
<!-- ... -->
  </attDef>
  <attList  org="choice">
    <attDef  ident="bar">
<!-- ... -->
    </attDef>
    <attDef  ident="baz">
<!-- ... -->
    </attDef>
  </attList>
</attList>
```

22.4.6 Element Classes

The element <classSpec> is used to document an element class or 'class', as defined in section *1.3. The TEI Class System*. It has the following components, additional to those already mentioned:

<classSpec> (class specification) contains reference information for a TEI element class; that is a group of elements which appear together in content models, or which share some common attribute, or both.

@**type** indicates whether this is a model class or an attribute class

<attList> contains documentation for all the attributes associated with this element, as a series of <attDef> elements.

A class specification does not list all of its members. Instead, its members declare that they belong to it by means of a <classes> element contained within the relevant <elementSpec>. This will contain a <memberOf> element for each class of which the relevant element is a member, supplying the name of the relevant class. For example, the <elementSpec> for the element <hi> contains the following:

```
<classes>
  <memberOf  key="model.hiLike"/>
</classes>
```

This indicates that the <hi> element is a member of the class with identifier model.hiLike. The <classSpec> element that documents this class contains the following declarations:

```
<classSpec  type="model"  ident="model.hiLike">
  <desc>groups phrase-level elements related to highlighting that have
     no specific semantics </desc>
  <classes>
    <memberOf  key="model.highlighted"/>
  </classes>
</classSpec>
```

which indicate that the class model.hiLike is actually a member (or subclass) of the class model.highlighted.

The attribute **type** is used to distinguish between 'model' and 'attribute' classes. In the case of attribute classes, the attributes provided by membership in the class are documented by an <attList> element contained within the <classSpec>. In the case of model classes, no further information is neeeded to define the class beyond its description, its identifier, and optionally any classes of which it is a member.

When a model class is referenced in the content model of an element (i.e. in the <content> of an <elementSpec>), its meaning will depend on the name used to reference the class.

If the reference simply takes the form of the class name, it is interpreted to mean an alternated list of all the current members of the class. For example, suppose that the members of the class model.hiLike are elements <hi>, <it>, and <bo>. Then a content model such as

```
<content>
  <rng:zeroOrMore>
    <rng:ref  name="model.hiLike"/>
  </rng:zeroOrMore>
</content>
```

would be equivalent to the explicit content model:

```
<content>
  <rng:zeroOrMore>
    <rng:choice>
      <rng:ref  name="hi"/>
      <rng:ref  name="it"/>
      <rng:ref  name="bo"/>
    </rng:choice>
  </rng:zeroOrMore>
</content>
```

(or, to use RELAX NG compact syntax, `(hi|it|bo)*`). However, a content model referencing the class as model.hiLike_sequence would be equivalent to the following explicit content model:

```
<content>
  <rng:zeroOrMore>
    <rng:ref  name="hi"/>
    <rng:ref  name="it"/>
    <rng:ref  name="bo"/>
  </rng:zeroOrMore>
</content>
```

(or, in RELAX NG compact syntax, `(hi,it,bo)*`.

The following suffixes, appended with an underscore, can be given to a class name when it is referenced in a content model:

alternation members of the class are alternatives

sequence members of the class are to be provided in sequence

sequenceOptional members of the class may be provided, in sequence, but are optional

sequenceOptionalRepeatable members of the class may be provided one or more times, in sequence, but are optional.

sequenceRepeatable members of the class must be provided one or more times, in sequence

Thus a reference to model.hiLike_sequenceOptional in a content model would be equivalent to:

```
<rng:zeroOrMore>
  <rng:optional>
    <rng:ref  name="hi"/>
  </rng:optional>
  <rng:optional>
    <rng:ref  name="it"/>
  </rng:optional>
  <rng:optional>
    <rng:ref  name="bo"/>
  </rng:optional>
</rng:zeroOrMore>
```

A reference to model.hiLike_sequenceRepeatable would however be equivalent to:

```
<rng:zeroOrMore>
  <rng:oneOrMore>
    <rng:ref  name="hi"/>
  </rng:oneOrMore>
  <rng:oneOrMore>
    <rng:ref  name="it"/>
  </rng:oneOrMore>
  <rng:oneOrMore>
    <rng:ref  name="bo"/>
  </rng:oneOrMore>
</rng:zeroOrMore>
```

and a reference to model.hiLike_sequenceOptionalRepeatable would be equivalent to:

```
<rng:zeroOrMore>
  <rng:zeroOrMore>
    <rng:ref  name="hi"/>
  </rng:zeroOrMore>
  <rng:zeroOrMore>
    <rng:ref  name="it"/>
  </rng:zeroOrMore>
  <rng:zeroOrMore>
    <rng:ref  name="bo"/>
  </rng:zeroOrMore>
</rng:zeroOrMore>
```

The 'sequence' in which members of a class appear in a content model when one of the sequence options is used is that in which the elements are declared.

In principal, all these possibilities are available to any element making reference to any class. The <classSpec> element defining the class may however limit the possibilities by means of its **generate** attribute, which can be used to say that this particular model may only be referenced in a content model with the suffixes it specifies. For example, if the <classSpec> for model.hiLike took the form **<classSpec ident="model.hiLike" generateOnly="sequence sequenceOptional">** then a content model referring to (say) model.hiLike_sequenceRepeatable would be regarded as invalid by an ODD processor.

When a <classSpec> contains an <attList> element, all the members of that class inherit the attributes specified by it. For example, the class att.interpLike defines a small set of attributes common to all elements which are members of that class: those attributes are listed by the <attList> element contained by the <classSpec> for att.interpLike. When processing the documentation elements for elements which are members of that class, an ODD processor is required to extend the <attList> (or equivalent) for such elements to include any attributes defined by the <classSpec> elements concerned. There is a single global attribute class, att.global, the membership of which may be expanded by some modules.

22.4.7 Pattern Documentation

The <macroSpec> element is used to document predefined strings or patterns not otherwise documented by the elements described in this chapter. Its chief uses are to provide systematic documentation of the parameter entities used within TEI DTD fragments and to describe common content models, but it may be used for any purpose. It has the following components additional to those already introduced:

<macroSpec> (macro specification) documents the function and implementation of a pattern.
- @**type** indicates which type of entity should be generated, when an ODD processor is generating a module using XML DTD syntax.

<remarks> contains any commentary or discussion about the usage of an element, attribute, class, or entity not otherwise documented within the containing element.

<stringVal> contains the intended expansion for the entity documented by a <macroSpec> element, enclosed by quotation marks.

22.5 Building a Schema

The specification elements, and several of their children, are all members of the att.identified class, from which they inherit the following attributes:

att.identified provides attributes for elements which can be referenced by means of a **key** attribute.
- @**ident** Supplies the identifier by which this element is referenced.
- @**predeclare** Says whether this object should be predeclared in the tei infrastructure module.
- @**module** Supplies the name of the module in which this object is to be defined.
- @**mode** specifies the effect of this declaration on its parent module.

These attributes are used by an ODD processor to determine how declarations are to be combined to form a schema or DTD, as further discussed in this section.

As noted above, a TEI schema is defined by a <schemaSpec> element containing an arbitrary mixture of explicit declarations for objects (i.e. elements, classes, patterns, or macro specifications) and references to other objects containing such declarations (i.e. references to specification groups, or to modules). A major purpose of this mechanism is to simplify the process of defining user customizations, by providing a formal method for the user to combine new declarations with existing ones, or to modify particular parts of existing declarations.

In the simplest case, a user-defined schema might simply combine all the declarations from two nominated modules:

```
<schemaSpec ident="example">
  <moduleRef key="teistructure"/>
  <moduleRef key="linking"/>
</schemaSpec>
```

An ODD processor, given such a document, would combine the declarations which belong to the named modules, and deliver the result as a schema of the requested type. It might also generate documentation for all and only the elements declared by those modules.

A schema might also include declarations for new elements, as in the following example:

```
<schemaSpec ident="example">
  <moduleRef key="teiheader"/>
  <moduleRef key="verse"/>
  <elementSpec ident="soundClip">
    <classes>
      <memberOf key="model.pPart.data"/>
    </classes>
  </elementSpec>
</schemaSpec>
```

A declaration for the element <soundClip>, which is not defined in the TEI scheme, will be added to the output schema. This element will also be added to the existing TEI class model.pPart.data, and will thus be available in TEI conformant documents.

A schema might also include re-declarations of existing elements, as in the following example:

```
<schemaSpec ident="example">
  <moduleRef key="teiheader"/>
  <moduleRef key="teistructure"/>
  <elementSpec ident="head" mode="change">
    <content>
      <rng:ref name="macro.xtext"/>
    </content>
  </elementSpec>
</schemaSpec>
```

The effect of this is to redefine the content model for the element <head> as plain text, by over-riding the <content> child of the selected <elementSpec>. The attribute specification `mode="change"` has the effect of over-riding only those children elements of the <elementSpec> which appear both in the original specification and in the new specification supplied above: <content> in this example. Note that if the value for **mode** were **replace**, the effect would be to replace all children elements of the original specification with the the children elements of the new specification, and thus (in this example) to delete all of them except <content>.

A schema may not contain more than two declarations for any given component. The value of the **mode** attribute is used to determine exactly how the second declaration (and its constituents) should be combined with the first. The following table summarizes how a processor should resolve duplicate declarations; the term identifiable refers to those elements which can have a **mode** attribute:

mode value	existing declaration	effect

add	no	add new declaration to schema; process its children in add mode
add	yes	raise error
replace	no	raise error
replace	yes	retain existing declaration; process new children in replace mode; ignore existing children
change	no	raise error
change	yes	process identifiable children according to their modes; process unidentifiable children in replace mode; retain existing children where no replacement or change is provided
delete	no	raise error
delete	yes	ignore existing declaration and its children

22.6 Combining TEI and Non-TEI Modules

In the simplest case, all that is needed to include a non-TEI module in a schema is to reference its RELAX NG source using the **url** attribute on <moduleRef>. The following specification, for example, creates a schema in which declarations from the non-TEI module svg11.rng (defining Standard Vector Graphics) are included. To avoid any risk of name clashes, the schema specifies that all TEI patterns generated should be prefixed by the string "TEI_".

```
<schemaSpec prefix="TEI_" ident="testsvg" start="TEI svg">
  <moduleRef key="header"/>
  <moduleRef key="core"/>
  <moduleRef key="tei"/>
  <moduleRef key="textstructure"/>
  <moduleRef url="svg11.rng"/>
</schemaSpec>
```

This specification generates a single schema which might be used to validate either a TEI document (with the root element <TEI>), or an SVG document (with a root element <svg:svg>), but would *not* validate a TEI document containing <svg:svg> or other elements from the SVG language. For that to be possible, the <svg:svg> element must become a member of a TEI model class (*1.3. The TEI Class System*), so that it may be referenced by other TEI elements. To achieve this, we modify the last <moduleRef> in the above example as follows:

```
<moduleRef url="svg11.rng">
  <content>
    <rng:define name="tei_model.graphicLike" combine="choice">
      <rng:ref name="svg"/>
    </rng:define>
  </content>
</moduleRef>
```

This states that when the declarations from the svg11.rng module are combined with those from the other modules, the declaration for the model class model.graphicLike in the TEI module should be extended to include the element <svg:svg> as an alternative. This has the effect that elements in the TEI scheme which define their content model in terms of that element class (notably <figure>) can now include it. A RELAX NG schema generated from such a specification can be used to validate documents in which the TEI <figure> element contains any valid SVG representation of a graphic, embedded within an <svg:svg> element.

22.7 Module for Documention Elements

The module described in this chapter makes available the following components:

Module tagdocs: Documentation of TEI modules

- Elements defined: altIdent att attDef attList attRef classSpec classes code content datatype defaultVal eg egXML elementSpec equiv exemplum gi ident listRef macroSpec memberOf moduleRef moduleSpec remarks schemaSpec specDesc specGrp specGrpRef specList stringVal tag val valDesc valItem valList
- Classes defined: att.identified
- Macros defined: macro.schemaPattern

The selection and combination of modules to form a TEI schema is described in *1.2. Defining a TEI Schema*.

The elements described in this chapter are all members of one of three classes: model.oddDecl, model.oddRef, or model.phrase.xml, with the exceptions of <schemaSpec> (a member of model.divPart) and both <eg> and <egXML> (members of model.common and model.egLike). All of these classes are declared along with the other general TEI classes, in the basic structure module documented in *1. The TEI Infrastructure*.

In addition, some elements are members of the att.identified class, which is documented in *22.5. Building a Schema* above, and make use of the `macro.schemaPattern` pattern, which is documented in *22.4.4. Element Specifications* above.

Chapter 23

Using the TEI

This section discusses some technical topics concerning the deployment of the TEI markup scheme documented elsewhere in these Guidelines. In section *23.2. Personalization and Customization* we discuss the scope and variety of the TEI customization mechanisms, distinguishing between 'clean' modifications, which result in a schema that supports a subset of the distinctions made in the full TEI system, on the one hand, from 'unclean' modifications, which result in a schema that does not have this property. In *23.3. Conformance* we define the notion of TEI Conformance, distinguishing between documents which are algorithmically TEI conformant ("TEI Conformable") from those which are intrinsically conformant ("TEI Conformant"); we also define the concept of a TEI extension. Since the ODD markup description language defined in chapter *22. Documentation Elements* is fundamental to the way conformance and customization are handled in the TEI system, these two definitional sections are followed by a section (*23.4. Implementation of an ODD System*) which describes the intended behaviour of an ODD processor.

23.1 Obtaining the TEI Schemas

As discussed in chapter *22. Documentation Elements*, the modules making up the TEI scheme are generated from a single set of XML source files. Schemas can be generated for TEI customizations in each of XML DTD language, W3C schema language, and RELAX NG schema language. In the body of the Guidelines, only the latter form is presented, using the compact syntax.

The TEI schemas and Guidelines are widely available over the Internet and elsewhere. The canonical home for the TEI source, the schema fragments generated from it, and example modifications, is the TEI repository at `http://tei.sf.net`; versions are also available in other formats, along with copies of the Guidelines and related materials, from the TEI web site at `http://www.tei-c.org`.

23.2 Personalization and Customization

These Guidelines provide an encoding scheme suitable for encoding a very wide range of texts, and capable of supporting a wide variety of applications. For this reason, the TEI scheme supports a variety of different approaches to solving similar problems, and also defines a much richer set of elements than is likely to be necessary in any given project. Furthermore, the TEI scheme may be extended in well-defined and documented ways for texts that cannot be conveniently or appropriately encoded using what is provided. For these reasons, it is almost impossible to use the TEI scheme without customizing or personalizing it in some way.

This chapter describes how the TEI encoding scheme may be customized, and should be read in conjunction with chapter *22. Documentation Elements*, which describes how a specific application of the TEI encoding scheme should be documented. The documentation system described in that chapter is, like the rest of the TEI scheme, independent of any particular schema or document type definition language.

Formally speaking, these Guidelines provide both syntactic rules about how elements and attributes may be used in valid documents and semantic recommendations about what interpretation should be attached to a given syntactic construct. In this sense, they provide both a document type definition and a document type declaration. More exactly, we may distinguish between the TEI abstract model, which defines a set of related concepts, and the TEI schema which defines a set of syntactic rules and constraints. Many (though not all) of the semantic recommendations are provided solely as informal descriptive prose, though some of them are also enforced by means of such constructs as datatypes (see *1.4.2. Datatype Macros*). Although the descriptions have been written with care, there will inevitably be cases where the intention of the contributors has not been conveyed with sufficient clarity to prevent users of the Guidelines from 'extending' them in the sense of attaching slightly variant semantics to them.

Beyond this unintentional semantic extension, some of the elements described can intentionally be used in a variety of ways; for example, the element <note> has an attribute **type** which can take on arbitrary string values, depending on how it is used in a document. A new type of 'note', therefore, requires no change in the existing model. On the other hand, for many applications, it may be desirable to constrain the possible values for the **type** attribute to a small set of possibilities. A schema modified in this way would no longer necessarily regard as valid the same set of documents as the corresponding unmodified TEI schema, but would remain faithful to the same conceptual model.

This section explains how the TEI scheme can be customized by suppressing elements, modifying classes of elements, adding elements, and renaming elements. Documents which validate against an application of the TEI scheme which has been customized in this way may or may not be considered 'TEI conformant', as further discussed in section *23.3. Conformance*.

The TEI scheme is designed to support modification and customization in a documented way that can be validated by an XML processor. This is achieved by writing a small TEI Conformant document, from which an appropriate processor can generate both human-readable documentation, and a schema expressed in a language such as RELAX NG or DTD. The mechanisms used to instantiate a TEI schema differ for different schema languages, and are therefore not defined here. In XML DTDs, for example, extensive use is made of parameter entities, while in RELAX NG schemas, extensive use is made of patterns. In either case, the names of elements and, wherever possible, their attributes and content models are defined indirectly. The syntax used to implement this indirection also varies with the schema language used, but the underlying constructs in the TEI abstract model are given the same names.

As further discussed in section *1. The TEI Infrastructure*, the TEI encoding scheme comprises a set of class and macro declarations, and a number of modules. Each module is made up of element and attribute declarations, and a schema is made by combining a particular set of modules together. In the absence of any other kind of personalization, when modules are combined together:

1. all the elements defined by the module (and described in the corresponding section of these Guidelines) are included in the schema;

2. each such element is identified by the canonical name given it in these Guidelines;

3. the content model of each such element is as defined by these Guidelines;

4. the names, datatypes, and permitted values declared for each attribute associated with each such element are as given in these Guidelines;

5. the elements comprising element classes and the meaning of macro declarations expressed in terms of element classes is determined by the particular combination of modules selected.

The TEI personalization mechanisms allow the user to control this behaviour as follows:

1. particular elements may be suppressed, removing them from any classes in which they are members, and also from any generated schema;
2. within certain limits, the name (generic identifier) associated with an element may be changed, without changing the semantic or syntactic properties of the element;
3. new elements may be added to an existing class, thus making them available in macros or content models defined in terms of those classes;
4. additional attributes, or attribute values, may be specified for an individual element or for classes of elements;
5. within certain limits, attributes, or attribute values, may also be removed either from an individual element or for classes of elements;
6. the characteristics inherited by one class from another class may be modified by modifying its class membership: all members of the class then inherit the changed characteristics;
7. the set of values legal for an attribute or attribute class may be constrained or relaxed by supplying or modifying a value list, or by modifying its datatype.

The modification mechanisms presented in this chapter are quite general, and may be used to make all the types of changes just listed.

The recommended way of implementing and documenting all such modifications is by means of the ODD system described in chapter *22. Documentation Elements*; in the remainder of this section we give specific examples to illustrate how that system may be applied. An ODD processor, such as the Roma application supported by the TEI, or any other comparable set of stylesheets will use the declarations provided by an ODD to generate appropriate sets of declarations in a specific schema language such as RELAX NG or the XML DTD language. We do not discuss in detail here how this should be done, since the details are schema language-specific; some background information about the methods used for XML DTD and RELAX NG schema generation is however provided in section *1.2. Defining a TEI Schema*. Several example ODD files are also provided as part of the standard TEI release: see further section *23.2.4. Examples of Modification* below.

23.2.1 Kinds of Modification

For ease of discussion, we distinguish the following different kinds of modification:

1. deletion of elements;
2. renaming of elements;
3. modification of content models;
4. modification of attribute and attribute-value lists;
5. modification of class membership;
6. addition of new elements.

Each of these is described in the following sections.

Each kind of modification changes the set of documents that will be considered valid according to the resulting schema. Any combination of unchanged TEI modules may be thought of as defining a certain set of documents. Each schema resulting from a modified combination of TEI modules will define a different set of documents. The set of documents valid according to the unmodified schema may or may not be properly

contained in the set of documents considered to be valid according to the modified schema. We use the term clean modification to describe a modification which regards as valid a subset of the documents considered valid by the same combination of TEI modules unmodified. Alternatively, the set of documents considered valid by the original schema might be disjoint from the set of documents considered valid by the modified schema, with neither being properly contained by the other. Modifications that have this result are called unclean modifications. Despite this terminology, unclean modifications are not particularly deprecated, and their use may often be vital to the success of a project. The concept is introduced solely to distinguish the effects of different kinds of modification.

Cleanliness can only be assessed with reference to elements in the TEI namespace.

23.2.1.1 Deletion of Elements

The simplest way to modify the supplied modules is to suppress one or more of the supplied elements. This is simply done by setting the **mode** attribute to **delete** on an <elementSpec> for the element concerned.

For example, if the <note> element is not to be used in a particular application, the schema specification concerned will contain a declaration like the following:

```
<elementSpec ident="note" module="core" mode="delete"/>
```

The **ident** attribute here supplies the canonical name of the element to be deleted, the **module** attribute identifies the module in which this element is declared, and the **mode** attribute specifies what is to be done with it. Note that the module name must be supplied explicitly, and that the schema specification in which this declaration appears must also contain a reference to the module itself. The full specification for a schema in which this modification is applied would thus be something like the following:

```
<schemaSpec ident="mySchema">
  <moduleRef key="core"/>
<!-- other modules used by this schema -->
  <elementSpec ident="note" module="core" mode="delete"/>
</schemaSpec>
```

In most cases, deletion is a clean modification, since most elements are optional. Documents that are valid with respect to the modified schema are also valid according to the unmodified schema. To say this another way, the set of documents matching the new schema is contained by the set of documents matching the original schema.

There are however some elements in the TEI scheme which have mandatory children; for example, the element <fileDesc> must contain both a <titleStmt> and a <sourceDesc>. A modification which deleted either of these would be unclean, because it would regard as valid documents that the unmodified schema would regard as invalid. Deleting one of the many optional children of <fileDesc> (<editionStmt> or <notesStmt> for example) would not have this effect, and would be a clean modification.

In general, whenever the element deleted by a modification is mandatory within the content model of some other (undeleted) element, the result is an unclean modification, and may also break the TEI abstract model (*23.3.3. Conformance to the TEI Abstract Model*). However, the parent of a mandatory child can be safely removed if it is itself optional.

To determine whether or not an element is mandatory in a given context, the user must inspect the content model of the element concerned. In most cases, content models are expressed in terms of model classes rather than elements; hence, removing an element will generally be a clean modification, since there will generally be other members of the class available. If a class is completely depopulated by a modification, then the cleanliness of the modification will depend upon whether or not the class reference is mandatory or optional, in the same way as for an individual element.

23.2.1.2 Renaming of Elements

Every element and other named markup construct in the TEI scheme has a canonical name, usually in the English language: this name is supplied as the value of the **ident** attribute on the <elementSpec>, <attDef>, <classSpec>, or <macroSpec> used to define it. The element or attribute declaration used within a schema generated from that specification may however be different, thus permitting schemas to be written using elements with generic identifiers from a different language, or otherwise modified. There may be many alternative identifiers for the same markup construct, and an ODD processor may choose which of them to use for a given purpose. Each such alternative name is supplied by means of an <altIdent> element within the specification element concerned.

For example, the following declaration converts <note> to <annotation>:

```
<elementSpec ident="note" module="core" mode="change">
  <altIdent>annotation</altIdent>
</elementSpec>
```

Note that the **mode** attribute on the <elementSpec> now takes the value **change** to indicate that those parts of the element specification not supplied are to be inherited from the standard definition. The content of the <altIdent> element will be used in place of the canonical **ident** value in the schema generated.

Renaming in this way is always a reversible modification. Although it is an inherently unclean modification (because the set of documents matched by the resulting schema is disjoint with the set matched by its unmodified equivalent), the process of converting any document in which elements have been renamed into an exactly equivalent document using canonical names is completely deterministic, requiring only access to the ODD in which the renaming has been specified. This assumes that the renamed elements used are not placed in the TEI namespace but either use a null namespace or some user-defined namespace, as further discussed in *23.2.2. Modification and Namespaces*; if this is not the case, care must be taken to avoid name collision between the new name and all existing TEI names. Furthermore, unclean modifications which do not specify a namespace are not conformant (see further *23.2. Personalization and Customization*)

The TEI provides a systematic set of renamings into languages other than English. These all use a language-specific namespace.

23.2.1.3 Modification of Content Models

The content model for an element in the TEI scheme is defined by means of a <content> element within the <elementSpec> which specifies it. As shown elsewhere in these Guidelines, the content model is defined using RELAX NG syntax, whether the resulting schema is expressed in RELAX NG or in some other schema language.

For example, the specification for the element <term> provided by the Guidelines contains a <content> element like the following:

```
<content>
  <rng:ref name="macro.phraseSeq"/>
</content>
```

This indicates that the content model contains declarations taken from the RELAX NG namespace, and that it consists of a reference to a pattern called macro.phraseSeq. Further examination shows that this pattern in turn expands to an optional repeatable alternation of text (`rng:text`) with references to three other classes (model.gLike, model.phrase, or model.global). For some particular application it might be preferable to insist that <term> elements should only contain plain text, excluding these other possibilities.[1] This could be achieved simply by supplying a specification for <term> like the following:

[1] Excluding model.gLike is generally inadvisable however, since without it the resulting schema has no way of referencing non-Unicode characters.

```
<elementSpec ident="term" module="core" mode="change">
  <content>
    <rng:text/>
  </content>
</elementSpec>
```

This is a clean modification which does not change the meaning of a TEI element; there is therefore no need to assign the element to some other namespace than that of the TEI, though it may be considered good practice; see further *23.2.2. Modification and Namespaces* below.

A change of this kind, which simplifies the possible content of an element by reducing its model to one of its existing components, is always clean, because the set of documents matched by the resulting schema is a subset of the set of documents which would have been matched by the unmodified schema.

Note that content models are generally defined (as far as possible) in terms of references to model classes, rather than to explicit elements. This means that the need to modify content models is greatly reduced: if an element is deleted or modified, for example, then the deletion or modification will be available for every content model which references that element via its class, as well as those which reference it explicitly. For this reason it is not (in general) good practice to replace class references by explicit element references, since this may have unintended side effects.

An unqualified reference to an element class within a content model generates a content model which is equivalent to an alternation of all the members of the class referenced. Thus, a content model which refers to the model class model.phrase will generate a content model in which any one of the members of that class is equally acceptable. It is also possible to reference predefined content model fragments based on classes, such as 'an optional repeatable alternation of all members of a class', 'a sequence containing no more than one of each member of the class', etc. as described further in *22.4.6. Element Classes*.

Content model changes which are not simple restrictions on an existing model should be undertaken with caution. The set of documents matching the schema which results from such changes is likely to be disjoint with the set of documents matching the unmodified schema, and such changes are therefore regarded as unclean. When content models are changed or extended, care should be taken to respect the existing semantics of the element concerned as stated in the Guidelines. For example, the element <l> is defined as containing a line of verse. It would not therefore make sense to redefine its content model so that it could also include members of the class model.pLike: such a modification although syntactically feasible would not be regarded as TEI conformant because it breaks the TEI abstract model.

23.2.1.4 Modification of Attribute and Attribute Value Lists

The attributes applicable to a given element may be specified in two ways: they may be given explicitly, by means of an <attList> element within the corresponding <elementSpec>, or they may be inherited from an attribute class, as specified in the <classes> element. To add a new attribute to an element, the schema builder should therefore first check to see whether this attribute is already defined by some existing attribute class. If it is, then the simplest method of adding it will be to make the element in question a member of that class, as further discussed below. If this is not possible, then a new <attDef> element must be added to the existing <attList> for the element in question.

Whichever method is adopted, the modification capabilities are the same as those available for elements. Attributes may be added or deleted from the list, using the **mode** attribute on <attDef> in the same way as on <elementSpec>. The 'content' of an attribute is defined by means of the <datatype>, <valList>, or <valDesc> elements within the <attDef> element. Any of these elements may be changed.

Suppose, for example, that we wish to add two attributes to the <eg> element (used to indicate examples in a text), **type** to characterize the example in some way, and **source** to indicate where the example comes from. A quick glance through the Guidelines indicates that the attribute class att.typed could be used to provide the

type attribute, but there is no comparable class which will provide a **source** attribute. The existing <eg> element in fact has no local attributes defined for it at all: we will therefore need to add not only an <attDef> element to define the new attribute, but also an <attList> to hold it.

We begin by adding the new **source** attribute:

```
<elementSpec ident="eg" module="tagdocs" mode="change">
 <attList>
  <attDef
     ident="source"
     mode="add"
     ns="http://www.example.org/ns/nonTEI">
   <desc>specifies the source of an example by pointing to a
        single bibliographic reference for it</desc>
   <datatype maxOccurs="1">
    <rng:ref name="data.pointer"/>
   </datatype>
  </attDef>
 </attList>
</elementSpec>
```

The value supplied for the **mode** attribute on the <attDef> element is **add**; if this attribute already existed on the element we are modifying this should generate an error, since a specification cannot have more than one attribute of the same name. If the attribute is already present, we can replace the whole of the existing declaration by supplying **replace** as the value for **mode**; alternatively, we can change some parts of an existing declaration only by supplying just the new parts, and setting **change** as the value for **mode**.

Because the new attribute is not defined by the TEI, we must specify a namespace for it on the <attDef>; see further *23.2.2. Modification and Namespaces*.

As noted above, adding the new **type** attribute involves changing this element's class membership; we therefore discuss that in the next section (*23.2.1.5. Class Modification*).

The canonical name for the new attribute is **source**, and is supplied on the **ident** attribute of the <attDef> element. In this simple example, we supply only a description and datatype for the new attribute; the former is given by the <desc> element, and the latter by the <datatype> element. (There are of course many other pieces of information which could be supplied, as documented in *22. Documentation Elements*). The content of the <datatype> element, like that of the <content> element, uses patterns from the RELAX NG namespace, in this case to select one of the predefined TEI datatypes (*1.4.2. Datatype Macros*).

It is often desirable to constrain the possible values for an attribute to a greater extent than is possible by simply supplying a TEI datatype for it. This facility is provided by the <valList> element, which can also appear as a child of the <attDef> element. Suppose for example that, rather than supplying them as pointers to a bibliography, all that we wish to indicate about the source of our examples is that each comes from one of three predefined sources, which we call A, B, and C. A declaration like the following might be appropriate:

```
<elementSpec ident="eg" module="tagdocs" mode="change">
 <attList>
  <attDef ident="source" ns="http://example.com/ns" mode="add">
   <desc>specifies the source of an example by supplying one of three
        predefined codes for it.</desc>
   <datatype maxOccurs="1">
    <rng:ref name="data.word"/>
   </datatype>
   <valList type="closed">
    <valItem ident="A">
```

```
        <desc>Examples taken from the A-list</desc>
       </valItem>
       <valItem ident="B">
        <desc>Examples taken from the B-list</desc>
       </valItem>
       <valItem ident="C">
        <desc>Examples taken from the C-list</desc>
       </valItem>
     </valList>
   </attDef>
  </attList>
</elementSpec>
```

The same technique may be used to replace or extend the <valList> supplied as part of any attribute in the TEI scheme.

Depending on the modification, the set of documents matched by a schema generated from an ODD modified in this way, may or may not be a subset of the set of documents matched by the unmodified schema. As such, it is difficult to tell in principle whether such modifications are intrinsically unclean.

23.2.1.5 Class Modification

The concept of element classes was introduced in *1.3.2. Model Classes*; an understanding of it is fundamental to successful use of the TEI scheme. As noted there, we distinguish model classes, the members of which all have structural similarity, from attribute classes, the members of which simply share a set of attributes.

The part of an element specification which determines its class membership is an element called <classes>. All classes to which the element belongs must be specified within this, using a <memberOf> element for each.

To add an element to a class in which it is not already a member, all that is needed is to supply a new <memberOf> element within the <classes> element for the element concerned. For example, to add an element to the att.typed class, we include a declaration like the following:

```
<elementSpec
   ident="eg"
   module="tagdocs"
   mode="change"
   ns="http://example.com/ns">
  <classes mode="change">
   <memberOf key="att.typed"/>
  </classes>
</elementSpec>
```

Any existing class memberships for the element being changed are not affected because the **mode** attribute of the <classes> element is set to **change** (rather than its default value of **replace**).

Consequently, in this case, the <eg> element retains its membership of the two classes (model.common and model.graphicLike) to which it already belongs.

Equally, to remove the attributes which an element inherits from its membership in some class, all that is needed is to remove the relevant <memberOf> element. For example, the element <term> defined in the core module is a member of two attribute classes, att.typed and att.declaring. It inherits the attributes **type** and **subtype** from the former, and the attribute **decls** from the latter. To remove the last of these attributes from this element, we need to remove it from that class:

```
<elementSpec
   ident="term"
   module="core"
   mode="change"
   ns="http://example.com/ns">
  <classes  mode="change">
    <memberOf  key="att.declaring"  mode="delete"/>
  </classes>
</elementSpec>
```

If the intention is to change the class membership of an element completely, rather than simply add or remove it to or from one or more classes, the value of **mode** attribute of <classes> can be set to **replace** (which is the default if no value is specified), indicating that the memberships indicated by its child <memberOf> elements are the only ones applicable. Thus the following declaration:

```
<elementSpec
   ident="term"
   module="core"
   mode="change"
   ns="http://example.com/ns">
  <classes  mode="replace">
    <memberOf  key="att.interpLike"/>
  </classes>
</elementSpec>
```

would have the effect of removing the element <term> from both its existing attribute classes, and adding it to the att.interpLike class.

If however the **mode** attribute is set to **change**, the implication is that the memberships indicated by its child <memberOf> elements are to be combined with the existing memberships for the element.

To change or remove attributes inherited from an attribute class for all members of the class (as opposed to specific members of that class), it is also possible to modify the class specification itself. For example, the class att.global defines several attributes which are available for all elements, notably **xml:id**, **xml:lang**, **rend**, and **rendition** among others. If we decide that we never wish to use the **rend** attribute, the simplest way of removing it is to supply a modified class specification for att.global as follows:

```
<classSpec  ident="att.global"  type="atts"  mode="change">
  <attList>
    <attDef  ident="rend"  mode="delete"/>
  </attList>
</classSpec>
```

Because the **mode** attribute on the <classSpec> defining the attributes inherited through membership of this class has the value **change**, any of its existing identifiable components not specified in the modification above will remain unchanged. The only effect will therefore be to delete the **rend** attribute from the class, and hence from all elements which are members of the class.

The classes used in the TEI scheme are further discussed in chapter *1. The TEI Infrastructure*. Note in particular that classes are themselves classified: the attributes inherited by a member of attribute class A may come to it directly from that class, or from another class of which A is itself a member. For example, the class att.global is itself a member of the classes att.global.linking and att.global.analytic. By default, these two classes are predefined as empty. However, if (for example) the linking module is included in a schema, a number of

attributes (**corresp**, **sameAs**, etc.) are defined as members of the att.global.linking class. All elements which are members of att.global will then inherit these new attributes (see further section *1.3.1. Attribute Classes*). A new attribute may thus be added to the global class in two ways: either by adding it to the <attList> defined within the class specification for att.global; or by defining a new attribute class, and changing the class membership of the att.global class to reference it.

Such global changes should be undertaken with caution: in general removing existing non-mandatory attributes from a class will always be a clean modification, in the same way as removing non-mandatory elements. Adding a new attribute to a class however can be a clean modification only if the new attribute is labelled as belonging to some namespace other than the TEI.

The same mechanisms are available for modification of model classes. Care should be taken when modifying the model class membership of existing elements since model class membership is what determines the content model of most elements in the TEI scheme, and a small change may have unintended consequences.

23.2.1.6 Addition of New Elements

To add a completely new element into a schema involves providing a complete element specification for it, the <classes> element of which includes a reference to at least one TEI model class. Without such a reference, the new element will not be referenced by the content model of any other TEI element, and will therefore be inaccessible within a TEI document.

For example, the three elements <bibl>, <biblFull>, and <biblStruct> are all defined as members of the class model.biblLike. To add a fourth member (say <myBibl>) to this class, we need to include in the <elementSpec> defining our new element a <memberOf> element which nominates the intended class:

```
<elementSpec ident="myBibl" mode="add" ns="http://www.example.com/ns/">
  <classes>
    <memberOf key="model.biblLike"/>
  </classes>
<!-- other parts of the new declaration here -->
</elementSpec>
```

The other parts of this declaration will typically include a description for the new element and information about its content model, its attributes, etc., as further described in *22. Documentation Elements*.

23.2.2 Modification and Namespaces

All the elements defined by the TEI scheme are labelled as belonging to a single namespace, maintained by the TEI and with the URI **http://www.tei-c.org/ns/1.0**.[2] Only elements which are unmodified or which have undergone a clean modification may use this namespace. In a TEI-conformant document, it is assumed that all attributes not explicitly labelled with a namespace (such as, for example **xml:id**) also belong to the TEI namespace, and are defined by the TEI.

This implies that any other modification (including a renaming or reversible modification) must either specify a different namespace or specify no namespace at all. The **ns** attribute is provided on elements <schemaSpec>, <elementSpec>, and <attDef> for this purpose.

Suppose, for example, that we wish to add a new attribute **topic** to the existing TEI element <p>. In the absence of namespace considerations, this would be an unclean modification, since <p> does not currently have such an attribute. The most appropriate action is to explicitly attach the new attribute to a new namespace by a declaration such as the following:

[2] This is not strictly the case, since the element <egXML> used to represent TEI examples has its own namespace, **http://www.tei-c.org/ns/Examples**; this is the only exception however.

```
<elementSpec ident="p" mode="change">
  <attList>
    <attDef ident="topic" mode="add" ns="http://www.example.org/ns/nonTEI">
      <desc>indicates the topic of a TEI paragraph</desc>
      <datatype>
<!-- ... -->
      </datatype>
    </attDef>
  </attList>
</elementSpec>
```

Document instances using a schema derived from this ODD can now indicate clearly the status of this attribute:

```
<div
   xmlns:my="http://www.example.org/ns/nonTEI">
<!-- ... -->
  <p n="12" my:topic="rabbits">Flopsy, Mopsy, Cottontail, and
    Peter...</p>
</div>
```

Since **topic** is explicitly labelled as belonging to something other than the TEI namespace, we regard the modification which introduced it as clean. A namespace-aware processor will be able to validate those elements in the TEI namespace against the unmodified schema.[3]

Similar methods may be used if a modification (clean or unclean) is made to the content model or some other aspect of an element, or if it declares a new element.

If the **ns** attribute is supplied on a <schemaSpec> element, it identifies the namespace applicable to all components of the schema being specified. Even if such a schema includes unmodified modules from the TEI namespace, the elements contained by such modules will now be regarded as belonging to the namespace specified on the <schemaSpec>. This can be useful if it is desired simply to avoid namespace processing. For example, the following schema specification results in a schema called noName which has no namespace, even though it comprises declarations from the TEI header module:

```
<schemaSpec ns="" ident="noName">
  <moduleRef key="header"/>
<!-- ... -->
</schemaSpec>
```

In addition to the TEI canonical namespace mentioned above, the TEI may also define namespaces for approved translations of the TEI scheme into other languages. These may be used as appropriate to indicate that a customization uses a standardized set of renamings. The namespace for such translations is the same as that for the canonical namespace, suffixed by the appropriate ISO language identifier (*vi.1 Language identification*). A schema specification using the Chinese translation, for example, would use the namespace http://www.tei-c.org/ns/1.0/zh

23.2.3 Documenting the Modification

The elements used to define a TEI customization (<schemaSpec>, <moduleRef>, <elementSpec>, etc.) will typically be used within a TEI document which supplies further information about the intended use of the

[3] Full namespace support does not exist in the DTD language, and therefore these techniques are available only to users of more modern schema languages such as RELAX NG or W3C Schema.

new schema, the meaning and application of any new or modified elements within it, and so on. This document will typically conform to a TEI (or other) schema which includes the module described in chapter *22. Documentation Elements*.[4]

Where the customization to be documented simply consists in a selection of modules, perhaps with some deletion of unwanted elements or attributes, the documentation need not specify anything further. Even here however it may be considered worthwhile to replace some of the semantic information provided by the unmodified TEI specification. For example, the <desc> element of an unmodified TEI <elementSpec> may describe an element in terms more general than appropriate to a particular project, or the <exemplum> elements within it may not illustrate the project's actual intended usage of the element, or the <remarks> element may contain discussions of matters irrelevant to the project. These elements may therefore be replaced or deleted within an <elementSpec> as necessary.

Radical revision is also possible. It is feasible to produce a modification in which the <teiHeader> or <text> elements are not required, or in which any other rule stated in these Guidelines is either not enforced or not enforceable. In fact, the mechanism, if used in an extreme way, permits replacement of all that the TEI has to say about every component of its scheme. Such revisions would result in documents that are not TEI conformant in even the broadest sense, and it is not intended that encoders use the mechanism in this way. We discuss exactly what is meant by the concept of TEI conformance in the next section, *23.3. Conformance*.

23.2.4 Examples of Modification

Several examples of customizations of the TEI are available as part of the standard release, within the directory Exemplars. They include the following:

tei_bare The schema generated from this customization is the minimum needed for TEI Conformance. It provides only a handful of elements.

tei_all The schema generated from this customization combines all available TEI modules, providing over 500 elements.

tei_allPlus The schema generated from this customization combines all available TEI modules with three other non-TEI vocabularies, specifically MathML, SVG, and XInclude.

It is unlikely that any project would wish to use any of these extremes unchanged. However, they form a useful starting point for customization, whether by removing modules from tei_all or tei_allPlus, or by replacing elements deleted from tei_bare. They also demonstrate how an ODD document may be constructed to provide a basic reference manual to accompany schemas generated from it.

Shortly after publication of the first edition of these Guidelines, as a demonstration of how the TEI encoding scheme might be adopted to meet 90% of the needs of 90% of the TEI user community, the TEI editors produced a brief tutorial defining one specific 'clean' modification of the TEI scheme, which they called TEI Lite. This tutorial and its associated DTD became very popular and are still available from the TEI web site at `http://www.tei-c.org/Guidelines/Customization/Lite/`. The tutorial and associated schema specification is also included as one of the Exemplars provided with TEI P5.

The Exemplars provided with TEI P5 also include a customization file from which a schema for the validation of other customization files may be generated. This ODD, called tei_odds, combines the four basic modules with the tagdocs, dictionaries, gaiji, linking, and figures modules as well as including the (non-TEI) module defining the RELAX NG language. This enables schemas derived from this customization file to validate examples contained within them in a number of ways, further described within the document.

[4] This module can be used to document any XML schema, and has indeed been used to document several non-TEI schemas.

23.3 Conformance

The notion of TEI Conformance is intended to assist in the description of the format and contents of a particular XML document instance or set of documents. It may be found useful in such situations as:

- interchange or integration of documents amongst different researchers or users;
- software specifications for TEI-aware processing tools;
- agreements for the deposit of texts in, and distribution of texts from, archives;
- specifying the form of documents to be produced by or for a given project.

It is not intended to provide any other evaluation, for example of scholarly merit, intellectual integrity, or value for money. A document may be of major intellectual importance and yet not be TEI Conformant; a TEI Conformant document may be of no scholarly value whatsoever.

In this section we explore several aspects of conformance, and in particular attempt to define how the term TEI Conformant should be used. The terminology defined here should be considered normative: users and implementors of the TEI Guidelines should use the phrases 'TEI Conformant', 'TEI Conformable', and 'TEI Extension' only in the senses given and with the usages described.

A document is TEI Conformant if it:

- is a well-formed XML document (*23.3.1. Well-formedness criterion*)
- can be validated against a TEI Schema, that is, a schema derived from the TEI Guidelines (*23.3.2. Validation Constraint*)
- conforms to the TEI Abstract Model (*23.3.3. Conformance to the TEI Abstract Model*)
- uses the TEI Namespace (and other namespaces where relevant) correctly (*23.3.4. Use of the TEI Namespace*)
- is documented by means of a TEI Conformant ODD file (*23.3.5. Documentation Constraint*) which refers to the TEI Guidelines

Each of these criteria is discussed in more detail below.

A document is said to be TEI Conformable if it is a well-formed XML document which can be transformed algorithmically and automatically into a TEI Conformant document as defined above without loss of information. Such a document may informally be described as TEI conformant; the terms algorithmically conformant or TEI Conformable are provided in order to distinguish documents exhibiting these kinds of conformance from others.

A document is said to use a TEI Extension if it is a well-formed XML document which is valid against a TEI Schema which contains additional distinctions, representing concepts not present in the TEI abstract model, and therefore not documented in these Guidelines. Such a document cannot, in general, be algorithmically conformant since it cannot be automatically transformed without loss of information. However, since one of the goals of the TEI is to support extensions and modifications, it should not be assumed that no TEI document can include extensions: an extension which is expressed by means of the recommended mechanisms is also a TEI document provided that those parts of it which are not extensions are TEI Conformant, or Conformable.

A TEI Conformant (or Conformable) document is said to follow TEI Recommended Practice if, wherever the Guidelines prefer one encoding practice to another, the preferred practice is used.

23.3.1 Well-formedness criterion

These Guidelines mandate the use of well-formed XML as representation format. Documents must conform to the World Wide Web Consortium recommendation of the *Extensible Markup Language (XML) 1.0 (Fourth Edition)* or successor editions found at http://www.w3.org/TR/xml/. Other ways of representing the concepts of the TEI abstract model are possible, and other representations may be considered appropriate for use

in particular situations (for example, for data capture, or project-internal processing). But such alternative representations are at best 'TEI Conformable', and cannot be considered in any way TEI Conformant.

Previous versions of these Guidelines used SGML as a representation format. With release P5, the only representation format supported by these Guidelines becomes valid XML; legacy documents in SGML format should therefore be converted using appropriate software.

A TEI Conformant document must use the TEI namespace, and therefore must also include an XML-conformant namespace declaration, as defined below (*23.3.4. Use of the TEI Namespace*).

The use of XML greatly reduces the need to consider hardware or software differences between processing environments when exchanging data. No special packing or interchange format is required for an XML document, beyond that defined by the W3C recommendations, and no special 'interchange' format is therefore proposed by these Guidelines. For discussion of encoding issues that may arise in the processing of special character sets or non-standard writing systems, see further chapter *vi Languages and Character Sets*.

In addition to the well-formedness criterion, the W3C defines the notion of a valid document, as being a well-formed document which matches a specific set of rules or syntactic constraints, defined by a schema. As noted above, TEI conformance implies that the schema used to determine validity of a given document should be derived from the present Guidelines, by means of an ODD which references and documents the schema fragments which the Guidelines define.

23.3.2 Validation Constraint

All TEI Conformant documents must validate against a schema file that has been derived from the published TEI Guidelines, combined and documented in the manner described in section *23.2. Personalization and Customization*. We call the formal output of this process a TEI Schema.

A TEI Schema may be expressed in any or all of the XML DTD language, W3C XML Schema, and RELAX NG (both compact and XML formats); the TEI does not mandate use of any particular schema language, only that this schema[5] should have been generated from a TEI ODD file that references the TEI Guidelines. Some of what is syntactically possible using the ODD formalism cannot be represented by all schema languages; and there are some features of some schema languages which have no counterpart in ODD. No single schema language fully captures all the constraints implied by conformance to the TEI abstract model. A document which is valid according to a TEI schema represented using one schema language may not be valid against the same schema expressed in other languages; in particular the DTD language does not fully support namespaces. Features which cannot be represented in all schema languages are documented in chapters *22. Documentation Elements* and *23.4. Implementation of an ODD System*.

As noted in section *23.2. Personalization and Customization*, many varieties of TEI schema are possible and not all of them are necessarily TEI Conformant; derivation from an ODD is a necessary but not a sufficient condition for TEI Conformance.

23.3.3 Conformance to the TEI Abstract Model

The TEI Abstract Model is the conceptual schema instantiated by the TEI Guidelines. These Guidelines define, both formally and informally, a set of abstract concepts such as 'paragraph' or 'heading', and their structural relationships, for example stating that 'paragraph's do not contain 'heading's. These Guidelines also define classes of elements, which have both semantic and structural properties in common. Those semantic and structural properties are also a part of the TEI abstract model; the class membership of an existing TEI element cannot therefore be changed without changing the model. Elements can however be removed from a class by deletion, and new non-TEI elements within their own namespaces can be added to existing TEI classes.

[5] Here and elsewhere we use the word *schema* to refer to any formal document grammar language, irrespective of the formalism used to represent it.

23.3.3.1 Semantic Constraints

It is an important condition of TEI conformance that elements defined in the TEI Guidelines as having one specific meaning should not be used with another. For example, the element <l> is defined in the TEI Guidelines as containing a line of verse. A schema in which it is redefined to mean a typographic line, or an ordered queue of objects of some kind, cannot therefore be TEI Conformant, whatever its other properties.

The semantics of elements defined in the TEI Guidelines are conveyed in a number of ways, ranging from formally verifiable datatypes to informal descriptive prose. In addition, a mapping between TEI elements and concepts in other conceptual models may be provided by the <equiv> element where this is available.

A schema which shares equivalent concepts to those of the TEI conceptual model may be mappable to the TEI Schema by means of such a mechanism. For example, the concept of paragraph, expressed in the TEI scheme by the <p> element is probably the same concept as that expressed in the Docbook scheme by the <para> element. In this respect (though not in others) a Docbook-conformant document might therefore be considered to be TEI Conformable. Such areas of overlap facilitate interoperability, because elements from one namespace may be readily integrated with those from another, but do not affect the definition of conformance.

A document is said to conform to the TEI Abstract Model if features for which an encoding is proposed by the TEI Guidelines are encoded within it using the markup and other syntactic properties specified by means of a valid TEI Conformant schema. That is, the abstract definition and markup structurally correspond to that in the TEI Guidelines even if the names of elements and attributes might not. Although it may be possible to transform a document following the TEI Abstract Model into a TEI Conformant document, it is not itself conformant.

23.3.3.2 Mandatory Components of a TEI Document

It is a long-standing requirement for any TEI Conformant document that it should contain a <teiHeader> element. To be more specific a TEI Conformant document must contain either:

- a single <teiHeader> element followed by a single <text> element, in that order; or
- in the case of a corpus or collection, a single overall <teiHeader> element followed by a series of <TEI> elements each with its own <teiHeader>

All <teiHeader> elements in a TEI Conformant document must include elements for:

Title Statement This should include the title of the TEI document expressed using a <titleStmt> element.

Publication Statement This should include the place and date of publication or distribution of the TEI document, expressed using the <publicationStmt> element.

Source Statement For a document derived from some previously existing document, this must include a bibliographic description of that source. For a document not so derived, this must include a brief statement that the document has no pre-existing source. In either case, this will be expressed using the <sourceDesc> element.

23.3.4 Use of the TEI Namespace

The Namespaces Recommendation of the W3C (Bray et al. (eds.) (2006)) provides a way for an XML document to combine markup from different vocabularies without risking name collision and consequent processing difficulties. While the scope of the TEI is large, there are many areas in which it makes no particular recommendation, or where it recommends that other defined markup schemes should be adopted, such as graphics or mathematics. It is also considered desirable that users of other markup schemes should be able to integrate documents using TEI markup with their own system. To meet these objectives without compromising the reliability of its encoding, a TEI Conformant document is required to make appropriate use of the TEI namespace.

Essentially all elements in a TEI Schema which represents concepts from the TEI abstract model belong to the TEI namespace, http://www.tei-c.org/ns/1.0, maintained by the TEI. A TEI Conformant document is required to declare the namespace for all the elements it contains whether these come from the TEI namespace or from other schemes.

A TEI Schema may be created which assigns TEI elements to some other namespace, or to no namespace at all. A document using such a schema must be regarded as a TEI extension and cannot be considered TEI Conformant, though it may be TEI Conformable. A document which places non-TEI elements or attributes within the TEI namespace cannot be TEI Conformant; such practices are strongly deprecated as they may lead to serious difficulties for processing or interchange.

23.3.5 Documentation Constraint

As noted in *23.3.2. Validation Constraint* above, a TEI Schema can only be generated from a TEI ODD, which also serves to document the semantics of the elements defined by it. A TEI Conformant document should therefore always be accompanied by (or refer to) a valid TEI ODD file specifying which modules, elements, classes, etc. are in use together with any modifications or renamings applied, and from which a TEI Schema can be generated to validate the document. The TEI supplies a number of predefined TEI Customization exemplar ODD files and the schemas already generated from them (see *23.2.4. Examples of Modification*), but most projects will typically need to customize the TEI beyond what these examples provide. It is assumed, for example, that most projects will customize the TEI scheme by removing those elements that are not needed for the texts they are encoding, and by providing further constraints on the attribute values and element content models the TEI provides. All such customizations must be specified by means of a valid TEI ODD file.

As different sorts of customization have different implications for the interchange and interoperability of TEI documents, it cannot be assumed that every customization will necessarily result in a schema that validates only TEI Conformant documents. The ODD language permits modifications which conflict with the TEI abstract model, even though observing this model is a requirement for TEI Conformance. The ODD language can in fact be used to describe many kinds of markup scheme, including schemes which have nothing to do with the TEI at all.

Equally, it is possible to construct a TEI Schema which is identical to that derived from a given TEI ODD file without using the ODD scheme. A schema can constructed simply by combining the predefined schema language fragments corresponding with the required set of TEI modules and other statements in the relevant schema language. The status of such a schema with respect to the tei_all schema cannot however be determined, in general; it may therefore be impossible to determine whether such a schema represents a clean modification or an extension. This is one reason for making the presence of a TEI ODD file a requirement for conformance.

23.3.6 Varieties of TEI Conformance

The conformance status of a given document may be assessed by answering the following questions, in the order indicated:

1. Is it a valid XML document, for which a TEI Schema exists? If not, then the document cannot be considered TEI Conformant in any sense.

2. Is the document accompanied by a TEI Conformant ODD specification describing its markup scheme and intended semantics? If not, then the document can only be considered TEI Conformant if it validates against a predefined TEI Schema and conforms to the TEI abstract model.

3. Does the markup in the document correctly represent the TEI abstract model? Though difficult to assess, this is essential to TEI conformance.

4. Does the document claim that all of its elements come from some namespace other than the TEI (or no namespace)? If so, the document cannot be TEI Conformant, though it may be TEI Conformable.

5. If the document claims to use the TEI namespace, in part or wholly, do the elements associated with that namespace in fact belong to it? If not, the document cannot be TEI Conformant; if so, and if all non-TEI elements and attributes are correctly associated with other namespaces, then the document may be TEI Conformant.

6. Is the document valid according to a schema made by combining all TEI modules as well as valid according to the schema derived from its associated ODD specification? If so, the document is TEI Conformant.

7. Is the document valid according to the schema derived from its associated ODD specification, but not according to tei_all? If so, the document uses a TEI extension.

8. Is it possible automatically to transform the document into a document which is valid according to tei_all, using only information supplied in the accompanying ODD and without loss of information? If so, the document is TEI Conformable.

In the following table, we examine more closely some specific, though imaginary, cases:

	A	B	C	D	E	F	G	H
Conforms to TEI abstract model	Y	N	Y	Y	?	Y	N	?
Valid ODD present	Y	Y	Y	Y	Y	Y	Y	N
Uses only non-TEI namespace(s) or none	N	N	N	N	Y	N	Y	N
Uses TEI and other namespaces correctly	Y	Y	N	Y	N	Y	N	Y
Document is valid as a subset of tei_all	Y	N	Y	N	N	Y	N	Y
Document can be converted automatically to a form which is valid as a subset of tei_all	Y	N	Y	N	N	Y	N	?

We assume firstly that each sample document assessed here is a well-formed XML document, and that it is valid against some schema.

The document in column A is TEI Conformant. Its tagging follows the TEI Abstract Model, both as regards syntactic constraints (its <l> elements appear within elements and not the reverse) and semantic constraints (its <l> elements appear to contain verse lines rather than typographic ones). It is accompanied by a valid ODD which documents exactly how it uses the TEI. All the TEI-defined elements and attributes in the document are placed in the TEI namespace. The schema against which it is valid is a 'clean' subset of the tei_all schema.

The document in column B is not a TEI document. Although it is accompanied by a valid TEI ODD, the resulting schema includes some 'unclean' modifications, and represents some concepts from the TEI Abstract Model using non-TEI elements; for example, it re-defines the content model of <p> to permit within it, and it includes an element <pageTrimming> which appears to have the same meaning as the existing TEI <fw> element, but the equivalence is not made explicit in the ODD. It uses the TEI namespace correctly to identify the TEI elements it contains, but the ODD does not contain enough information automatically to convert its non-TEI elements into TEI equivalents.

The document in column C is TEI Conformable. It is almost the same as the document in column A, except that the names of the elements used are not those specified by the TEI namespace. Because the ODD accompanying it contains an exact mapping for each element name (using the <altIdent> element) and there are no name conflicts, it is possible to make an automatic conversion of this document.

The document in column D is a TEI Extension. It combines elements from its own namespace with unmodified TEI elements in the TEI namespace. Its usage of TEI elements conforms to the TEI Abstract Model. Its ODD defines a new <blort> element which has no exact TEI equivalent, but which is assigned to an existing TEI class; consequently its schema is not a clean subset of tei_all. If the associated ODD provided a way of mapping this element to an existing TEI element, then this would be TEI Conformable.

The document in column E is superficially similar to document D, but because it does not use any namespace declarations (or, equivalently, it assigns unmodified TEI elements to its own namespace), it may contain name collisions; there is no way of knowing whether a <p> within it is the same as the TEI's <p> or has some other meaning. The accompanying ODD file may be used to provide the human reader with information about equivalently named elements in the TEI namespace, and hence to determine whether the document is valid with respect to the TEI Abstract Model but this is not an automatable process. In particular, cases of apparent conflict (for example use of an element <p> to represent a concept not in the TEI Abstract Model but in the Abstract Model of some other system, whose namespace has been removed as well) cannot be reliably resolved. By our current definition therefore, this is not a TEI document.

The document in column F is TEI Conformable. The difference between it and that in column D is that the new element <blort> which is used in this document is a specialisation of an existing TEI element, and the ODD in which it is defined specifies the mapping (a <my:blort> may be automatically converted to a **<tei:seg type="blort">**, for example). For this to work, however, the <blort> must observe the same syntactic constraints as the <seg>; if it does not, this would also be a case of TEI Extension.

The document in column G is not a TEI document. Its structure is fully documented by a valid TEI ODD, but it does not claim to represent the TEI Abstract Model, does not use the TEI namespace, and is not intended to validate against any TEI schema.

The document in column H is very like that in column A, but it lacks an accompanying ODD. Instead, the schema used to validate it is produced simply by combining TEI schema fragments in the same way as an ODD processor would, given the ODD. If the resulting schema is a clean subset of tei_all, such a document is indistinguishable from a TEI Conformant one, but there is no way of determining (without inspection) whether this is the case if any modification or extension has been applied. Its status is therefore, like that of Text E, impossible to determine.

23.4 Implementation of an ODD System

This chapter specifies how a processing system may take advantage of the markup specification elements documented in chapter *22. Documentation Elements* of these Guidelines in order to produce project specific user documentation, schemas in one or more schema languages, and validation tools for other processors.

The specifications in this chapter are illustrative but not normative. Its function is to further illustrate the intended scope and application of the elements documented in chapter *22. Documentation Elements*, since it is believed that these may have application beyond the areas directly addressed by the TEI.

An ODD processing system has to accomplish two main tasks. A set of selections, deletions, changes, and additions supplied by an ODD customization (as described in *23.2. Personalization and Customization*) must first be merged with the published TEI P5 ODD specifications. Next, the resulting unified ODD must be processed to produce the desired outputs.

An ODD processor is not required to do these two stages in sequence, but that may well be the simplest approach; the ODD processing tools currently provided by the TEI Consortium, which are also used to process the source of these Guidelines, adopt this approach.

23.4.1 Making a Unified ODD

An ODD customization must contain a single <schemaSpec> element, which defines the schema to be constructed.

<schemaSpec> (schema specification) generates a TEI-conformant schema and documentation for it.

- @**ns** (namespace) specifies the default namespace (if any) applicable to components of the schema.
- @**start** specifies entry points to the schema, i.e. which elements are allowed to be used as the root of documents conforming to it.
- @**prefix** specifies a prefix which will be appended to all patterns relating to TEI elements. This allows for external schemas to be mixed in which have elements of the same names as the TEI.
- @**targetLang** (target language) specifies which language to use when creating the objects in a schema if names for elements or attributes are available in more than one language, .
- @**docLang** (documentation language) specifies which languages to use when creating documentation if the description for an element, attribute, class or macro is available in more than one language, .

Amongst other attributes inherited from the att.identified class, this element also carries a required **ident** attribute. This provides a name for the generated schema, which other components of the processing system may use to refer to the schema being generated, e.g. in issuing error messages or as part of the generated output schema file or files. The **ns** attribute may be used to specify the default namespace within which elements valid against the resulting schema belong, as discussed in *23.2.2. Modification and Namespaces*.

The <schemaSpec> element contains an unordered series of specialized elements, each of which is of one of the following four types:

specifications elements from the class model.oddDecl (by default <elementSpec>, <classSpec>, <moduleSpec>, and <macroSpec>); these must have a **mode** attribute which determines how they will be processed.[6] If the value of **mode** is **add**, then the object is simply copied to the output, but if it is **change**, **delete**, or **replace**, then it will be looked at by other parts of the process.

references to specifications <specGrpRef> elements refer to <specGrp> elements that occur elsewhere in this, or another, document. A <specGrp> element, in turn, groups together a set of ODD specifications (among other things, including further <specGrpRef> elements). The use of <specGrp> and <specGrpRef> permits the ODD markup to occur at the points in documentation where they are discussed, rather than all inside <schemaSpec>. The **target** attribute of any <specGrpRef> should be followed, and the <elementSpec>, <classSpec>, and <macroSpec>, elements in the corresponding <specGrp> should be processed as described in the previous item; <specGrpRef> elements should be processed as described here.

references to TEI Modules <moduleRef> elements with **key** attributes refer to components of the TEI. The value of the **key** attribute matches the **ident** attribute of the <moduleSpec> element defining a TEI module. The **key** must be dereferenced by some means, such as reading an XML file with the TEI ODD specification (either from the local hard drive or off the Web), or looking up the reference in an XML database (again, locally or remotely); whatever means is used, it should return a stream of XML containing the element, class, and macro specifications collected together in the specified module. These specification elements are then processed in the same way as if they had been supplied directly within the <schemaSpec> being processed.

references to external modules a <moduleRef> element may also refer to a compatible external module by means of its **url** attribute; the content of such modules, which must be available in the RELAX NG XML syntax, are passed directly and without modification to the output schema when that is created.

[6] An ODD processor should recognize as erroneous such obvious inconsistencies as an attempt to include an <elementSpec> in **add** mode for an element which is already present in an imported module.

Each object obtained from the TEI ODD specification using <moduleRef> by means of the **key** attribute must be checked against objects in the customization <schemaSpec> according to the following rules:

1. if there is an object in the ODD customization with the same value for the **ident** attribute, and a **mode** value of **delete**, then the object from the module is ignored;
2. if there is an object in the ODD customization with the same value for the **ident** attribute, and a **mode** value of **replace**, then the object from the module is ignored, and the one from the ODD customization is used in its place;
3. if there is an object in the ODD customization with the same value for the **ident** attribute, and a **mode** value of **change**, then the two objects must be merged, as described below;
4. if there is an object in the ODD customization with the same value for the **ident** attribute, and a **mode** value of **add**, then an error condition should be raised;
5. otherwise, the object from the module is copied to the result.

To merge two objects with the same **ident**, their component attributes and child elements must be looked at recursively. Each component may fall into one of the following four categories:

1. Some components may occur only once within the merged object (for example attributes, and <altIdent>, <content>, or <classes> elements). If such a component is found in the ODD customization, it will be copied to the output; if it is not found there, but is present in the TEI ODD specification, then that will be copied to the output.
2. Some components are grouping objects (<attList>, <valList>, for example); these are always copied to the output, and their children are then processed following the rules given in this list.
3. Some components are 'identifiable': this means that they are members of the att.identified class from which they inherit the **ident** attribute; examples include <attDef> and <valItem>. A components of this type will be processed according to its **mode** attribute, following the rules given in this list.
4. Some components may occur multiple times, but are neither grouped nor identifiable. Examples include the members of model.glossLike such as <equiv>, <desc>, <gloss>, the <exemplum>, <remarks>, <listRef>, <datatype> or <defaultVal> elements. These should be copied from both the TEI ODD specification and the ODD customization, and all occurrences included in the output.

A special problem arises with elements which are members of attribute classes, as they are permitted to override attributes inherited from a class. For example, consider this simple modification:

```
<elementSpec ident="p">
  <classes>
    <memberOf key="att.typed"/>
  </classes>
  <content>
<!--…-->
  </content>
</elementSpec>
```

The effect of its membership in the att.typed class is to provide <p> with a **type** attribute and a **subtype** attribute. If we wish <p>*not* to have **subtype**, we could extend the customization in our schema as follows:

```
<elementSpec ident="p">
  <classes>
    <memberOf key="att.typed"/>
  </classes>
  <content>
<!--… -->
  </content>
  <attList>
    <attDef ident="subtype" mode="delete"/>
  </attList>
</elementSpec>
```

This means that when **<memberOf key="att.typed"/>** is processed, that class is looked up, each attribute which it defines is examined in turn, and the customization is searched for an override. If the modification is of the attribute class itself, work proceeds as usual; if, however, the modification is at the element level, the class reference is deleted and a series of <attRef> elements is added to the element, one for each attribute inherited from the class. Since attribute classes can themselves be members of other attribute classes, membership must be followed recursively.

The effect of the concatenation of unidentifiable components should be considered carefully. An original may have

```
<elementSpec ident="p">
  <desc>marks paragraphs in prose.</desc>
<!--…-->
</elementSpec>
```

which would usefully be extended with this:

```
<elementSpec ident="p" mode="change">
  <desc xml:lang="es">marca párrafos en prosa.</desc>
<!--…-->
</elementSpec>
```

to provide an alternate description in another language. Nothing prevents the user from supplying <desc> several times in the same language, and subsequent applications will have to decide what that may mean.

Similar considerations apply to multiple example elements, though these are less likely to cause problems in documentation. Note that existing examples can only be deleted by supplying a completely new <elementSpec> in **replace** mode, since the <exemplum> element is not identifiable.

In the processing of the content models of elements and the content of macros, deleted elements may require special attention.[7] A content model like this:

```
<elementSpec ident="person">
<!--…-->
  <content>
    <rng:choice>
      <rng:oneOrMore>
```

[7] The carthago program behind the Pizza Chef application, written by Michael Sperberg-McQueen for TEI P3 and P4, went to very great efforts to get this right. The XSLT transformations used by the P5 Roma application are not as sophisticated, partly because the RELAX NG language is more forgiving than DTDs.

```
      <rng:ref  name="model.pLike"/>
     </rng:oneOrMore>
     <rng:zeroOrMore>
      <rng:choice>
       <rng:ref  name="model.personPart"/>
       <rng:ref  name="model.global"/>
      </rng:choice>
     </rng:zeroOrMore>
    </rng:choice>
  </content>
<!--…-->
</elementSpec>
```

requires no special treatment because everything is expressed in terms of model classes; if deletions result in model.personPart having no members, then model.global is left as the only child of <rng:choice>. An ODD processor may or may not elect to simplify the resulting choice between nothing and model.global by removing the wrapper <rng:choice> element. However, such simplification may be considerably more complex in the general case (if for example the <rng:choice> is itself inside an <rng:zeroOrMore> inside a <rng:group>), and an ODD processor is therefore likely to be more successful in carrying out such simplification as a distinct stage during processing of ODD sources.

If an element refers directly to an element child, like this:

```
<elementSpec  ident="figure">
<!--…-->
  <content>
   <rng:zeroOrMore>
    <rng:choice>
     <rng:ref  name="model.pLike"/>
     <rng:ref  name="model.global"/>
     <rng:ref  name="figure"/>
     <rng:ref  name="figDesc"/>
     <rng:ref  name="model.graphicLike"/>
     <rng:ref  name="model.headLike"/>
    </rng:choice>
   </rng:zeroOrMore>
  </content>
<!--…-->
</elementSpec>
```

and <figDesc> has been deleted,[8] it will be necessary to remove that reference, or the resulting schema will be invalid. Surrounding constructs, such as a <rng:zeroOrMore> (which cannot be empty), may also have to be removed.

The result of the work carried out should be a new <schemaSpec> which contains a complete and internally consistent set of element, class, and macro specifications, possibly also including <moduleRef> elements with **url** attributes identifying external modules.

23.4.2 Generating Schemas

Assuming that any modifications have been resolved, as outlined in the previous section, making a schema is now a four stage process:

[8] Note that deletion of required elements will cause the schema specification to acccept as valid documents which cannot be TEI Conformant, since they no longer conform to the TEI abstract model; conformance topics are addressed in more detail in *23.3. Conformance*.

1. all datatype and other macro specifications must be collected together and declared at the start of the output schema;
2. all classes must be declared in the right order (since some classes reference others, the order is significant);
3. all elements are declared;
4. any <moduleRef> elements with a **url** attribute identifying an external schema must be processed.

Working in this order gives the best chance of successfully supporting all the schema languages. However, there are a number of obstacles to overcome along the way.

An ODD processor may use any desired schema language or languages for its schema output. The TEI ODD specification uses RELAX NG to express content models, and is therefore biased towards this language. However, the current TEI ODD processing system is capable of producing schema output in the three main schema languages, as follows:

- A RELAX NG (XML) schema is generated by creating wrappers around the content models taken directly from the ODD specification; a version re-expressed in the RELAX NG compact syntax is generated using James Clark's trang application.
- A DTD schema is generated by converting the RELAX NG content models to DTD language, often simplifying it to allow for the less-sophisticated output language.
- A W3C Schema schema is created by generating a RELAX NG schema and then using James Clark's trang application.

Note that the method used to generate W3C Schema means that a processor must ensure that the RELAX NG it generates follows the subset which trang is able to translate properly (see further below) — this may involve simple trial and error.

Other projects may decide to follow a different route, perhaps implementing a direct ODD to W3C Schema translator.

Secondly, it is possible to create two rather different styles of schema. On the one hand, the schema can try to maintain all the flexibility of ODD by using the facilities of the schema language for parameterization; on the other, it can remove all customization features and produce a flat result which is not suitable for further manipulation. The TEI project currently generates both styles of schema; the first as a set of schema fragments in DTD and RELAX NG languages, which can be included as modules in other schemas, and customized further; the second as the output from a processor such as Roma, in which many of the parameterization features have been removed.

The difference between the schema styles may be illustrated by considering this ODD specification:

```
<elementSpec module="drama" ident="performance">
<!-- ... -->
  <classes>
    <memberOf key="model.frontPart.drama"/>
  </classes>
  <content>
    <rng:group>
      <rng:zeroOrMore>
        <rng:choice>
          <rng:ref name="model.divTop"/>
          <rng:ref name="model.global"/>
        </rng:choice>
```

```
     </rng:zeroOrMore>
     <rng:oneOrMore>
       <rng:group>
         <rng:ref  name="model.common"/>
       </rng:group>
       <rng:zeroOrMore>
         <rng:ref  name="model.global"/>
       </rng:zeroOrMore>
     </rng:oneOrMore>
     <rng:zeroOrMore>
       <rng:ref  name="model.divBottom"/>
       <rng:zeroOrMore>
         <rng:ref  name="model.global"/>
       </rng:zeroOrMore>
     </rng:zeroOrMore>
   </rng:group>
  </content>
<!-- ... -->
</elementSpec>
```

A simple rendering to RELAX NG produces this:

```
    performance =
    element performance {
    (model.divTop | model.global)*,
    (model.common, model.global*)+,
    (model.divBottom, model.global*)*
    att.global.attribute.xmlspace,
    att.global.attribute.xmlid,
    att.global.attribute.n,
    att.global.attribute.xmllang,
    att.global.attribute.rend,
    att.global.attribute.xmlbase,
    att.global.linking.attribute.corresp,
    att.global.linking.attribute.synch,
    att.global.linking.attribute.sameAs,
    att.global.linking.attribute.copyOf,
    att.global.linking.attribute.next,
    att.global.linking.attribute.prev,
    att.global.linking.attribute.exclude,
    att.global.linking.attribute.select
    }
```

In the above, a subsequent redefinition of the attribute class (such as att.global) would have no effect, since references to such classes have been expanded to reference their constituent attributes.

The equivalent parameterized version might look this this:

```
    performance =
    element performance { performance.content, performance.attributes }
    performance.content =
    (model.divTop | model.global)*,
    (model.common, model.global*)+,
    (model.divBottom, model.global*)*
    performance.attributes = att.global.attributes, empty
```

Here, the attribute class att.global is provided via an explicit reference (`att.global.attributes`), and can therefore be redefined. Moreover, the attributes are separated from the content model, allowing either to be overridden.

In the remainder of these chapter, the terms simple schema and parameterized schema are used to distinguish the two schema types. An ODD processor is not required to support both, though the simple schema output is generally preferable for most applications.

Thirdly, the problem of missing components must be resolved. For example, consider this (fictitious) model for <sp>:

```
<elementSpec ident="sp">
<!--…-->
  <content>
    <rng:zeroOrMore>
      <rng:ref name="model.global"/>
    </rng:zeroOrMore>
    <rng:optional>
      <rng:ref name="speaker"/>
      <rng:zeroOrMore>
        <rng:ref name="model.global"/>
      </rng:zeroOrMore>
    </rng:optional>
  </content>
<!--…-->
</elementSpec>
```

This proposes anything from the global model class, followed by some <speaker> elements, followed by anything from the model.global class. What happens if <speaker> is removed? The following would result:

```
<elementSpec ident="sp">
<!--…-->
  <content>
    <rng:zeroOrMore>
      <rng:ref name="model.global"/>
    </rng:zeroOrMore>
    <rng:zeroOrMore>
      <rng:ref name="model.global"/>
    </rng:zeroOrMore>
  </content>
<!--…-->
</elementSpec>
```

which is illegal in DTD and W3C schema languages, since for a given member of model.global it is impossible to be sure which rule is being used. This situation is not detected when RELAX NG is used, since the language is able to cope with non-deterministic content models of this kind and does not require that only a single rule be used.

Finally, an application will need to have some method of associating the schema with document instances that use it. The TEI does not mandate any particular method of doing this, since different schema languages and processors vary considerably in their requirements. ODD processors may wish to build in support for some of the methods for associating a document instance with a schema. The TEI does not mandate any particular method, but does suggest that those which are already part of XML (the DOCTYPE declaration for DTDs) and W3C Schema (the **xsi:schemaLocation** attribute) be supported where possible.

In order for the **xsi:schemaLocation** attribute to be valid when a document is validated against either a DTD or a RELAX NG schema, ODD processors may wish to add declarations for this attribute and its namespace to the root element, even though these are not part of the TEI *per se*. For DTDs this means adding

```
xsi:schemaLocation CDATA #IMPLIED xmlns:xsi CDATA #FIXED
'http://www.w3.org/2001/XMLSchema-instance'
```

to the list of attributes on the root element, which permits the non-namespace-aware DTD language to recognize the `xsi:schemaLocation` notation. For RELAX NG, the namespace and attribute would be declared in the usual way:

```
namespace xsi =
"http://www.w3.org/2001/XMLSchema-instance"
```

and

```
attribute
xsi:schemaLocation { list { data.namespace, data.pointer }+ }
```

inside the root element declaration.

Note that declaration of the **xsi:schemaLocation** attribute in a W3C Schema schema is not permitted. Therefore, if W3C Schemas are being generated by converting the RELAX NG schema (for example, with trang), it may be necessary to perform that conversion prior to adding the **xsi:schemaLocation** declaration to the RELAX NG.

It is recognised that this is an unsatisfactory solution, but it permits users to take advantage of the W3C Schema facility for indicating a schema, while still permitting documents to be validated using DTD and RELAX NG processors without any conflict.

23.4.3 Names and Documentation in Generated Schemas

When processing class, element, or macro specifications, there are three general rules:

1. If a RELAX NG pattern or DTD parameter entity is being created, its name is the value of the corresponding **ident** attribute, prefixed by the value of any **prefix** attribute on <schemaSpec>. This allows for elements from an external schema to be mixed in without risk of name clashes, since all TEI elements can be given a distinctive prefix such as **tei_**. Thus

```
<schemaSpec ident="test" prefix="tei_">
  <elementSpec ident="sp">
<!--...-->
  </elementSpec>
</schemaSpec>
```

may generate a RELAX NG (compact syntax) pattern like this:

```
tei_sp = element sp { ... }
```

References to these patterns (or, in DTDs, parameter entities) also need to be prefixed with the same value.

2. If an element or attribute is being created, its default name is the value of the **ident** attribute, but if there is an <altIdent> child, its content is used instead.

3. Where appropriate, the documentation strings in <gloss> and <desc> should be copied into the generated schema. If there is only one occurrence of either of these elements, it should be used regardless, but if there are several, local processing rules will need to be applied. For example, if there are several with different values of **xml:lang**, a locale indication in the processing environment might be used to decide which to use. For example,

```
<elementSpec module="core" ident="head">
  <equiv/>
  <gloss>heading</gloss>
  <gloss xml:lang="fr">en-tête</gloss>
  <gloss xml:lang="es">encabezamiento</gloss>
  <gloss xml:lang="it">titolo</gloss>
<!-- ... -->
</elementSpec>
```

might generate a RELAX NG schema fragment like the following, if the locale is determined to be French:

```
head =
## en-tête
element head { head.content, head.attributes }
```

Alternatively, a selection might be made on the basis of the value of the **version** attribute which these elements carry as members of the att.translatable class.

In addition, there are three conventions about naming patterns relating to classes; ODD processors need not follow them, but those reading the schemas generated by the TEI project will find it necessary to understand them:

1. when a pattern for an attribute class is created, it is named after the attribute class identifier (as above) suffixed by `.attributes` (e.g. `att.editLike.attributes`);

2. when a pattern for an attribute is created, it is named after the attribute class identifer (as above) suffixed by `.attribute.` and then the identifier of the attribute (e.g. `att.editLike.attribute.resp`);

3. when a parameterized schema is created, each element generates patterns for its attributes and its contents separately, suffixing respectively `.attributes` and `.contents` to the element name.

23.4.4 Making a RELAX NG Schema

To create a RELAX NG schema, the processor processes every <macroSpec>, <classSpec>, and <elementSpec> in turn, creating a RELAX NG pattern for each, using the naming conventions listed above. The order of declaration is not important, and a processor may well sort them into alphabetical order of identifier.

A complete RELAX NG schema must have an <rng:start> element defining which elements can occur as the root of a document. The ODD <schemaSpec> has an optional **start** attribute, containing one or more element names, which can be used to construct the <rng:start>.

23.4.4.1 Macros

An ODD macro generates a corresponding RELAX NG pattern simply by copying the body of the <content> element. Thus

```
<macroSpec module="tei" type="pe" ident="macro.phraseSeq">
  <content>
    <rng:zeroOrMore>
      <rng:choice>
        <rng:text/>
        <rng:ref name="model.gLike"/>
        <rng:ref name="model.phrase"/>
        <rng:ref name="model.global"/>
      </rng:choice>
    </rng:zeroOrMore>
  </content>
</macroSpec>
```

produces

```
<rng:define name="macro.phraseSeq">
  <rng:zeroOrMore>
    <rng:choice>
      <rng:text/>
      <rng:ref name="model.gLike"/>
      <rng:ref name="model.phrase"/>
      <rng:ref name="model.global"/>
    </rng:choice>
  </rng:zeroOrMore>
</rng:define>
```

Although some versions of these Guidelines show the RELAX NG output in the compact syntax, both the content of the <content> element and the unified ODD specification generated by the TEI ODD processing software always store RELAX NG in the more verbose XML syntax. However, the two formats are interchangeable.

23.4.4.2 Classes

An ODD model class reference generates a RELAX NG pattern definition listing all the members of the class present in the ODD in alternation. So this example

```
<classSpec module="tei" type="model" ident="model.measureLike">
<!-- ... -->
</classSpec>
```

may produce, for a given customization:

```
<rng:define name="model.measureLike">
  <rng:choice>
    <rng:ref name="num"/>
    <rng:ref name="measure"/>
    <rng:ref name="measureGrp"/>
  </rng:choice>
</rng:define>
```

if the elements <num>, <measure>, and <measureGrp> are included. Depending on the value of the **generate** attribute on the <classSpec>, it may also generate a set of sequences as well as alternation patterns. Thus we may also generate the sequence, sequenceOptional, sequenceRepeatable, and sequenceOptionalRepeatable patterns:

```
<rng:define  name="model.measureLike_sequence">
  <rng:ref  name="num"/>
  <rng:ref  name="measure"/>
  <rng:ref  name="measureGrp"/>
</rng:define>
<rng:define  name="model.measureLike_sequenceOptional">
  <rng:optional>
    <rng:ref  name="num"/>
  </rng:optional>
  <rng:optional>
    <rng:ref  name="measure"/>
  </rng:optional>
  <rng:optional>
    <rng:ref  name="measureGrp"/>
  </rng:optional>
</rng:define>
<rng:define
    name="model.measureLike_sequenceOptionalRepeatable">
  <rng:zeroOrMore>
    <rng:ref  name="num"/>
  </rng:zeroOrMore>
  <rng:zeroOrMore>
    <rng:ref  name="measure"/>
  </rng:zeroOrMore>
  <rng:zeroOrMore>
    <rng:ref  name="measureGrp"/>
  </rng:zeroOrMore>
</rng:define>
<rng:define  name="model.measureLike_sequenceRepeatable">
  <rng:oneOrMore>
    <rng:ref  name="num"/>
  </rng:oneOrMore>
  <rng:oneOrMore>
    <rng:ref  name="measure"/>
  </rng:oneOrMore>
  <rng:oneOrMore>
    <rng:ref  name="measureGrp"/>
  </rng:oneOrMore>
</rng:define>
```

where the pattern name is created by appending an underscore and the name of the generation sequence to the class name.

Attribute classes work by producing a pattern containing definitions of the appropriate attributes. So

```
<classSpec  module="verse"  type="atts"  ident="att.enjamb">
  <attList>
    <attDef  ident="enjamb"  usage="opt">
      <equiv/>
      <desc>indicates whether the end of a verse line is marked by enjambement.</desc>
      <datatype>
        <rng:ref  name="data.enumerated"/>
```

```
     </datatype>
     <valList  type="open">
       <valItem  ident="no">
         <equiv/>
         <desc>the line is end-stopped
         </desc>
       </valItem>
       <valItem  ident="yes">
         <equiv/>
         <desc>the line in question runs on into the next
         </desc>
       </valItem>
       <valItem  ident="weak">
         <equiv/>
         <desc>the line is weakly enjambed
         </desc>
       </valItem>
       <valItem  ident="strong">
         <equiv/>
         <desc>the line is strongly enjambed</desc>
       </valItem>
     </valList>
   </attDef>
  </attList>
</classSpec>
```

produces

```
 <define xmlns="http://relaxng.org/ns/structure/1.0" name="att.enjamb.attributes">
<ref name="att.enjamb.attribute.enjamb"/>
<empty/>
</define>
<define xmlns="http://relaxng.org/ns/structure/1.0" name="att.enjamb.attribute.enjamb">
<optional>
<attribute name="enjamb">
<a:documentation xmlns:a="http://relaxng.org/ns/compatibility/annotations/1.0">(enjambement) indicates whether
the end of a verse line is marked by enjambement.
Sample values include: 1] no; 2] yes; 3] weak; 4] strong</a:documentation>
<ref name="data.enumerated"/>
</attribute>
</optional>
</define>
```

Since the processor may have expanded the attribute classes already, separate patterns are generated for each attribute in the class as well as one for the class itself. This allows an element to refer directly to a member of a class. Notice that the <desc> element is used to add an <a:documentation> element to the schema, which some editors use to provide help during composition. The <desc> elements in the <valList> are used to create the human-readable sentence

> Sample values include: 1] no; 2] yes; 3] weak; 4] strong

Naturally, this behaviour is not mandatory; and other ODD processors may create documentation in other ways, or ignore those parts of the ODD specifications when creating schemas.

An individual attribute consists of an <rng:attribute> with a **name** attribute derived according to the naming rules described above (*23.4.3. Names and Documentation in Generated Schemas*). In addition, the ODD model supports a <defaultVal>, which is transformed to a **defaultValue** attribute in the namespace http://relaxng.org/ns/compatibility/annotations/1.0 attribute on the <rng:attribute>. The body of the attribute is taken from the <datatype> child, unless there is a supporting <valList> with a **type** value of **closed**. In that case an <rng:choice> is created, listing the allowed values. Thus the following attribute definition

```
<attDef ident="full" usage="opt">
  <defaultVal>yes</defaultVal>
  <valList type="closed">
    <valItem ident="yes">
      <desc>the name component is spelled out in full.</desc>
    </valItem>
    <valItem ident="abb">
      <gloss>abbreviated</gloss>
      <desc>the name component is given in an abbreviated form.</desc>
    </valItem>
    <valItem ident="init">
      <gloss>initial letter</gloss>
      <desc>the name component is indicated only by one initial.</desc>
    </valItem>
  </valList>
</attDef>
```

may generate this RELAX NG code:

```
<rng:define name="att.full">
  <rng:optional>
    <rng:attribute name="full" a:defaultValue="yes">
      <rng:choice>
        <rng:value>yes</rng:value>
        <a:documentation>the name component is spelled out in full.
        </a:documentation>
        <rng:value>abb</rng:value>
        <a:documentation>the name component is given in an abbreviated form.
        </a:documentation>
        <rng:value>init</rng:value>
        <a:documentation>the name component is indicated only by one initial.
        </a:documentation>
      </rng:choice>
    </rng:attribute>
  </rng:optional>
</rng:define>
```

Note the use of the http://relaxng.org/ns/compatibility/annotations/1.0 namespace to provide default values and documentation.

23.4.4.3 Elements

An <elementSpec> produces a RELAX NG specification in two parts; firstly, it must generate an <rng:define> pattern by which other elements can refer to it, and then it must generate an <rng:element> with the content model and attributes. It may be convenient to make two separate patterns, one for the element's attributes and one for its content model.

The content model is created simply by copying the body of the <content> element; the attributes are processed in the same way as those from attribute classes, described above.

23.4.5 Making a DTD

Generation of DTDs largely follows the same pattern as RELAX NG generation, with one important exception — **the order of declaration matters**. A DTD may not refer to an entity which has not yet been declared. Since both macros and classes generate DTD parameter entities, the TEI Guidelines are constructed so that they can be declared in the right order. A processor must therefore work in the following order:

1. declare all model classes which have a **predeclare** value of **true**
2. declare all macros which have a **predeclare** value of **true**
3. declare all other classes
4. declare the modules (if DTD fragments are being constructed)
5. declare any remaining macros
6. declare the elements and their attributes

Let us consider a complete example, a simple element with no attributes of its own:

```
<elementSpec  module="namesdates"  ident="faith">
  <desc>specifies the faith, religion, or belief set of a person.</desc>
  <classes>
    <memberOf  key="model.persTraitLike"/>
    <memberOf  key="att.editLike"/>
    <memberOf  key="att.datable"/>
  </classes>
  <content>
    <rng:ref  name="macro.phraseSeq"/>
  </content>
</elementSpec>
```

If DTD fragments are being generated (for use as described in *23.4.7. Using TEI Parameterized Schema Fragments*), this will result in the following:

```
    <!ENTITY % faith 'INCLUDE' >
    <![ %faith; [
    <!--doc:specifies the faith,  religion, or belief set of a person.  -->
    <!ELEMENT %n.faith; %om.RR; %macro.phraseSeq;>
    <!ATTLIST %n.faith; xmlns CDATA "http://www.tei-c.org/ns/1.0">
    <!ATTLIST %n.faith;
    %att.global.attributes;
    %att.editLike.attributes;
    %att.datable.attributes; >
    ]]>
```

Here the whole stanza is contained in a marked section (for use as described in *23.4.7.2. Inclusion and Exclusion of Elements*), the element name is parameterized (see *23.4.7.3. Changing the Names of Generic Identifiers*), and the class attributes are entity references derived from the <memberOf> records in <classes>. Note the additional attribute which provides a default **xmlns** declaration for the element; the effect of this is that if the document is processed by a DTD-aware XML processor, the namespace declaration will be present automatically without the document author even being aware of it.

A simpler rendition for a flattened DTD generated from a customization will result in the following, with no containing marked section, and no parameterized name:

```
<!ELEMENT faith %macro.phraseSeq;>
<!ATTLIST faith xmlns CDATA "http://www.tei-c.org/ns/1.0">
<!ATTLIST faith
%att.global.attribute.xmlspace;
%att.global.attribute.xmlid;
%att.global.attribute.n;
%att.global.attribute.xmllang;
%att.global.attribute.rend;
%att.global.attribute.xmlbase;
%att.global.linking.attribute.corresp;
%att.global.linking.attribute.synch;
%att.global.linking.attribute.sameAs;
%att.global.linking.attribute.copyOf;
%att.global.linking.attribute.next;
%att.global.linking.attribute.prev;
%att.global.linking.attribute.exclude;
%att.global.linking.attribute.select;
%att.editLike.attribute.cert;
%att.editLike.attribute.resp;
%att.editLike.attribute.evidence;
%att.datable.w3c.attribute.period;
%att.datable.w3c.attribute.when;
%att.datable.w3c.attribute.notBefore;
%att.datable.w3c.attribute.notAfter;
%att.datable.w3c.attribute.from;
%att.datable.w3c.attribute.to;>
```

Here the attributes from classes have been expanded into individual entity references.

23.4.6 Generating Documentation

In Donald Knuth's literate programming terminology (Knuth (1992)), the previous sections have dealt with the tangle process; to generate documentation, we now turn to the weave process.

An ODD customization may consist largely of general documentation and examples, requiring no ODD-specific processing. It will normally however also contain a <schemaSpec> element and possibly some <specGrp> fragments.

The generated documentation may be of two forms. On the one hand, we may document the customization itself, that is, only those elements (etc.) which differ in their specification from that provided by the TEI reference documentation. Alternatively, we may generate reference documentation for the complete subset of the TEI which results from applying the customization. The TEI Roma tools take the latter approach, and operate on the result of the first stage processing described in *23.4.1. Making a Unified ODD*.

Generating reference documentation for <elementSpec>, <classSpec>, and <macroSpec> elements is largely dependent on the design of the preferred output. Some applications may, for example, want to turn all names of objects into hyperlinks, show lists of class members, or present lists of attributes as tables, lists, or inline prose. Another technique implemented in these Guidelines is to show lists of potential 'parents' for each element, by tracing which other elements have them as possible members of their content models.

One model of display on a web page is shown in Figure 6, Example reference documentation for <faith>, corresponding to the <faith> element shown in section *23.4.5. Making a DTD*.

23.4.7 Using TEI Parameterized Schema Fragments

The TEI parameterized DTD and RELAX NG fragments make use of parameter entities and patterns for several purposes. In this section we describe their interface for the user. In general we recommend use of ODD instead of this technique.



<faith>

Home | Table of Contents
Element catalogue

<**faith**> specifies the faith, religion, or belief set of a person.

Module	namesdates — 13 Names, Dates, People, and Places
Attributes	[att.editLike att.datable]

Declaration	Compact to XML format
	element faith
	{
	att.global.attributes,
	att.editLike.attributes,
	att.datable.w3c.attributes,
	att.datable.iso.attributes,
	macro.phraseSeq
	}

Example	<faith>protestant</faith>
Contained by	model.persTraitLike

Figure 23.1: Example reference documentation for <faith>

23.4.7.1 Selection of Modules

Special-purpose parameter entities are used to specify which modules are to be combined into a TEI DTD. They take the form **TEI.xxxxx** where xxxx is the name of the module as given in table the table on p. 2 in *1.1. TEI Modules*. For example, the parameter entity TEI.linking is used to define whether or not to include the module linking. All such parameter entities are declared by default with the value **IGNORE**: to select a module, therefore, the encoder declares the appropriate parameter entities with the value **INCLUDE**.

For XML DTD fragments, note that some modules generate two DTD fragments: for example the analysis module generates fragments called analysis-decl and analysis. This is because the declarations they contain are needed at different points in the creation of an XML DTD.

The parameter entity named for the module is used as the keyword controlling a conditional marked section in the DTD fragment generated by the tei module. The declarations for each DTD fragment constituting the module are contained within such marked sections. For example, the parameter entity TEI.linking appears twice in tei.dtd, once for the linking-decl schema fragment:

```
    <!ENTITY % TEI.linking 'IGNORE' >
    <![%TEI.linking;[
    <!ENTITY % file.linking-decl PUBLIC '-//TEI P5//ENTITIES Linking, Segmentation, and Alignment//EN' 'linking-
decl.dtd' >
    %file.linking-decl;
    ]] >
```

and once for the linking schema fragment:

```
    <![%TEI.linking;[
    <!ENTITY % file.linking PUBLIC '-//TEI P5//ELEMENTS Linking, Segmentation, and Alignment//EN' 'linking.dtd'
>
    %file.linking;
    ]] >
```

If TEI.linking has its default value of IGNORE, neither declaration has any effect. If however it has the value INCLUDE, then the content of each marked section is acted upon: the parameter entities file.linking and file.linking-decl are referenced, which has the effect of embedding the content of the files they represent at the appropriate point in the DTD.

The RELAX NG schema fragments can be combined in a wrapper schema using the standard mechanism of <rng:include> in that language.

23.4.7.2 Inclusion and Exclusion of Elements

The TEI DTD fragments also use marked sections and parameter entity references to allow users to exclude the definitions of individual elements, in order either to make the elements illegal in a document or to allow the element to be redefined. The parameter entities used for this purpose have exactly the same name as the generic identifier of the element concerned. The default definition for these parameter entities is **INCLUDE** but they may be changed to **IGNORE** in order to exclude the standard element and attribute definition list declarations from the DTD.

The declarations for the element <p>, for example, are preceded by a definition for a parameter entity with the name p and contained within a marked section whose keyword is given as `%p;`:

```
<!ENTITY % p 'INCLUDE' >
<![ %p; [
<!-- element and attribute list declaration for p here -->
]]
```

These parameter entities are defined immediately preceding the element whose declarations they control; because their names are completely regular, they are not documented further.

To define a DTD in which the element <p> is excluded therefore, the entity p needs to be redefined as **IGNORE** by ensuring that a declaration such as

```
<!ENTITY % p 'IGNORE' >
```

is added earlier in the DTD than the default (see further *23.4.7.4. Embedding Local Modifications (DTD only)*).

Similarly, in the parameterized RELAX NG schemas, every element is defined by a pattern named after the element. To undefine an element therefore all that is necessary is to add a declaration like the following:

```
p = notAllowed
```

23.4.7.3 Changing the Names of Generic Identifiers

In the TEI DTD fragments, elements are not referred to directly by their generic identifiers; instead, the DTD fragments refer to parameter entities which expand to the standard generic identifiers. This allows users to rename elements by redefining the appropriate parameter entity. Parameter entities used for this purpose are formed by taking the standard generic identifier of the element and attaching the string **n.** as a prefix. Thus the standard generic identifiers for paragraphs, notes, and quotations, <p>, <note>, and <persName> are defined by declarations of the following form:

```
<!ENTITY % n.p "p">
<!ENTITY % n.note "note">
<!ENTITY % n.persName "persName">
```

Note that since all names are case-sensitive, the specific mix of uppercase and lowercase letters in the standard generic identifier must be preserved in the entity name.

These declarations are generated by an ODD processor when TEI DTD fragments are created.

In the RELAX NG schemas, all elements are normally defined using a pattern with the same name as the element (as described in *23.4.3. Names and Documentation in Generated Schemas*): for example

```
abbr = element abbr { abbr.content, abbr.attributes }
```

The easiest way of renaming the element is thus simply to rewrite the pattern with a different element name; any references use the pattern, not the element, name.

```
abbr = element abbrev { abbr.content, abbr.attributes }
```

More complex revisions, such as redefining the content of the element (defined by the pattern abbr.content) or its attributes (defined by the pattern abbr.attributes) can be accomplished in a similar way, using the features of the RELAX NG language. The recommended method of carrying out such modifications is however to use the ODD language as further described in section *22. Documentation Elements*.

23.4.7.4 Embedding Local Modifications (DTD only)

Any local modifications to a DTD (i.e. changes to a schema other than simple inclusion or exclusion of modules) are made by declarations stored in one of two local extension files, one containing modifications to the TEI parameter entities, and the other new or changed declarations of elements and their attributes. Entity declarations must be made which associate the names of these two files with the appropriate parameter entity so that the declarations they contain can be embedded within the TEI DTD at an appropriate point.

The following entities are referred to by the main tei.dtd file to embed portions of the TEI DTD fragments or locally developed extensions.

TEI.extensions.ent identifies a local file containing extensions to the TEI parameter entities

TEI.extensions.dtd identifies a local file containing extensions to the TEI module

For example, if the relevant files are called project.ent and project.dtd, then declarations like the following would be appropriate:

```
<!ENTITY % TEI.extensions.ent SYSTEM 'project.ent' >
<!ENTITY % TEI.extensions.dtd SYSTEM 'project.dtd' >
```

When an entity is declared more than once, the first declaration is binding and the others are ignored. The local modifications to parameter entities should therefore be handled before the standard parameter entities themselves are declared in tei.dtd. The entity TEI.extensions.ent is referred to before any TEI declarations are handled, to allow the user's declarations to take priority. If the user does not provide a TEI.extensions.ent entity, the entity will be expanded to the empty string.

For example the encoder might wish to add two phrase-level elements <it> and <bd>, perhaps as synonyms for **<hi rend='italics'>** and **<hi rend='bold'>**. As described in chapter *23.2. Personalization and Customization*, this involves two distinct steps: one to define the new elements, and the other to ensure that they are placed into the TEI document structure at the right place.

Creating the new declarations is done in the same way for user-defined elements as for any other; the same parameter entities need to be defined so that they may be referenced by other elements. The content models

of these new elements may also reference other parameter entities, which is why they need to be declared after other declarations.

The second step involves modifying the element class to which the new elements should be attached. This requires that the parameter entity macro.phraseSeq should be modified to include the generic identifiers for the new elements we wish to create. The declaration for each modifiable parameter entity in the DTD includes a reference to an additional parameter entity with the same name prefixed by an `x.`; these entities are declared by default as the null string. However, in the file containing local declarations they may be redeclared to include references to the new class members:

```
<!ENTITY % x.macro.phraseSeq 'it | bd
|'>
```

and this declaration will take precedence over the default when the declaration for macro.phraseSeq is evaluated.

Appendix A

Model Classes

---

model.addrPart groups elements such as names or postal codes which may appear as part of a postal address.

Module tei — *1. The TEI Infrastructure*

Used by address

Members model.nameLike [model.nameLike.agent [name orgName persName] model.offsetLike [geogFeat offset] model.persNamePart [addName forename genName nameLink roleName surname] model.placeStateLike [model.placeNamePart [bloc country district geogName placeName region settlement] state] lang rs] addrLine postBox postCode street

---

model.addressLike groups elements used to represent a postal or e-mail address.

Module tei — *1. The TEI Infrastructure*

Used by location model.pPart.data

Members address affiliation email

---

model.applicationLike groups elements used to record application-specific information about a document in its header.

Module header — *2. The TEI Header*

Used by appInfo

Members application

---

model.biblLike groups elements containing a bibliographic description.

Module tei — *1. The TEI Infrastructure*

Used by broadcast cit climate event listBibl location org place population relatedItem scriptStmt sourceDesc state taxonomy terrain trait model.inter

Members bibl biblFull biblStruct msDesc

---

model.biblPart groups elements which represent components of a bibliographic description.

Module tei — *1. The TEI Infrastructure*

Used by bibl

Members model.imprintPart [biblScope distributor pubPlace publisher] model.respLike [author editor respStmt] edition extent idno meeting msIdentifier relatedItem series

---

model.castItemPart groups component elements of an entry in a cast list, such as dramatic role or actor's name.

Module tei — *1. The TEI Infrastructure*

Used by castItem

Members actor role roleDesc

---

model.catDescPart groups component elements of the TEI Header Category Description.

Module tei — *1. The TEI Infrastructure*

Used by catDesc

Members textDesc

---

model.choicePart groups elements (other than <choice> itself) which can be used within a <choice> alternation.

Module tei — *1. The TEI Infrastructure*

Used by choice

Members abbr am corr ex expan orig reg seg sic unclear

---

model.common groups common chunk- and inter-level elements.

Module tei — *1. The TEI Infrastructure*

Used by argument body castList div div1 div2 div3 div4 div5 div6 div7 epigraph epilogue performance postscript prologue set

Members model.divPart [model.divPart.spoken [u] model.lLike [l] model.pLike [ab p] eTree floatingText forest forestGrp graph lg schemaSpec sp tree] model.entryLike [entry entryFree superEntry] model.inter [model.biblLike [bibl biblFull biblStruct msDesc] model.egLike [eg egXML] model.labelLike [desc label] model.listLike [list listBibl listNym listOrg listPerson listPlace listWit] model.oddDecl [classSpec elementSpec listRef macroSpec moduleSpec specGrp] model.oddRef [moduleRef specGrpRef] model.qLike [model.quoteLike [cit quote] q said] model.stageLike [camera caption move sound stage tech view] castList figure table]

Note This class defines the set of chunk- and inter-level elements; it is used in many content models, including those for textual divisions.

---

model.dateLike groups elements containing temporal expressions.

Module tei — *1. The TEI Infrastructure*
Used by imprint setting model.pPart.data model.recordingPart
Members date time

---

model.div1Like groups top-level structural divisions.
Module tei — *1. The TEI Infrastructure*
Used by back body front
Members div1

---

model.div2Like groups second-level structural divisions.
Module tei — *1. The TEI Infrastructure*
Used by div1
Members div2

---

model.div3Like groups third-level structural divisions.
Module tei — *1. The TEI Infrastructure*
Used by div2
Members div3

---

model.div4Like groups fourth-level structural divisions.
Module tei — *1. The TEI Infrastructure*
Used by div3
Members div4

---

model.div5Like groups fifth-level structural divisions.
Module tei — *1. The TEI Infrastructure*
Used by div4
Members div5

---

model.div6Like groups sixth-level structural divisions.
Module tei — *1. The TEI Infrastructure*
Used by div5
Members div6

---

model.div7Like groups seventh-level structural divisions.
Module tei — *1. The TEI Infrastructure*

Used by div6
Members div7

---

model.divBottom groups elements appearing at the end of a text division.

Module tei — *1. The TEI Infrastructure*
Used by body div div1 div2 div3 div4 div5 div6 div7 epilogue group lg list performance prologue
Members model.divBottomPart [closer postscript signed trailer] model.divWrapper [argument byline dateline docAuthor docDate epigraph meeting]

---

model.divBottomPart groups elements which can occur only at the end of a text division.

Module tei — *1. The TEI Infrastructure*
Used by back front model.divBottom
Members closer postscript signed trailer

---

model.divGenLike groups elements used to represent a structural division which is generated rather than explicitly present in the source.

Module tei — *1. The TEI Infrastructure*
Used by body div div1 div2 div3 div4 div5 div6
Members divGen

---

model.divLike groups elements used to represent un-numbered generic structural divisions.

Module tei — *1. The TEI Infrastructure*
Used by back body div front
Members div

---

model.divPart groups paragraph-level elements appearing directly within divisions.

Module tei — *1. The TEI Infrastructure*
Used by specGrp macro.specialPara model.common
Members model.divPart.spoken [u] model.lLike [l] model.pLike [ab p] eTree floatingText forest forestGrp graph lg schemaSpec sp tree
Note Note that this element class does not include members of the model.inter class, which can appear either within or between paragraph-level items.

---

model.divPart.spoken groups elements structurally analogous to paragraphs within spoken texts.

Module spoken — *8. Transcriptions of Speech*
Used by model.divPart
Members u

Note Spoken texts may be structured in many ways; elements in this class are typically larger units such as turns or utterances.

---

model.divTop groups elements appearing at the beginning of a text division.

Module tei — *1. The TEI Infrastructure*

Used by body castList div div1 div2 div3 div4 div5 div6 div7 epilogue group lg list performance prologue

Members model.divTopPart [model.headLike [head] opener salute] model.divWrapper [argument byline dateline docAuthor docDate epigraph meeting]

---

model.divTopPart groups elements which can occur only at the beginning of a text division.

Module tei — *1. The TEI Infrastructure*

Used by model.divTop

Members model.headLike [head] opener salute

---

model.divWrapper groups elements which can appear at either top or bottom of a textual division.

Module tei — *1. The TEI Infrastructure*

Used by model.divTop model.divBottom

Members argument byline dateline docAuthor docDate epigraph meeting

---

model.editorialDeclPart groups elements which may be used inside <editorialDecl> and appear multiple times.

Module header — *2. The TEI Header*

Used by editorialDecl

Members correction hyphenation interpretation normalization quotation segmentation stdVals

---

model.egLike groups elements containing examples or illustrations.

Module tei — *1. The TEI Infrastructure*

Used by figure model.inter

Members eg egXML

---

model.emphLike groups phrase-level elements which are typographically distinct and to which a specific function can be attributed.

Module tei — *1. The TEI Infrastructure*

Used by model.highlighted model.limitedPhrase

Members code distinct emph foreign gloss ident mentioned soCalled term title

model.encodingPart groups elements which may be used inside <encodingDesc> and appear multiple times.

Module header — *2. The TEI Header*

Used by encodingDesc

Members appInfo charDecl classDecl editorialDecl fsdDecl geoDecl metDecl projectDesc refsDecl samplingDecl tagsDecl variantEncoding

---

model.entryLike groups elements structurally analogous to paragraphs within dictionaries.

Module dictionaries — *9. Dictionaries*

Used by model.common

Members entry entryFree superEntry

---

model.entryPart groups elements appearing at any level within a dictionary entry.

Module tei — *1. The TEI Infrastructure*

Used by cit dictScrap entryFree nym

Members case colloc def etym form gen gramGrp hom hyph iType lbl mood number orth per pos pron re sense stress subc superEntry syll tns usg xr

---

model.entryPart.top groups high level elements within a structured dictionary entry

Module tei — *1. The TEI Infrastructure*

Used by entry hom re sense

Members cit def dictScrap etym form gramGrp re usg xr

Note Members of this class typically contain related parts of a dictionary entry which form a coherent subdivision, for example a particular sense, homonym, etc.

---

model.featureVal groups elements which represent feature values in feature structures.

Module tei — *1. The TEI Infrastructure*

Used by f fvLib if vAlt vDefault vLabel vMerge vNot vRange

Members model.featureVal.complex [fs vColl vMerge vNot] model.featureVal.single [binary default numeric string symbol vAlt vLabel]

---

model.featureVal.complex groups elements which express complex feature values in feature structures.

Module tei — *1. The TEI Infrastructure*

Used by model.featureVal

Members fs vColl vMerge vNot

model.featureVal.single group elements used to represent atomic feature values in feature structures.

Module tei — *1. The TEI Infrastructure*

Used by vColl model.featureVal

Members binary default numeric string symbol vAlt vLabel

---

model.formPart groups elements allowed within a <form> element in a dictionary.

Module dictionaries — *9. Dictionaries*

Used by form

Members model.gramPart [model.morphLike [case gen gram iType mood number per tns] colloc gramGrp lbl pos subc usg] form hyph orth pron syll

---

model.frontPart groups elements which appear at the level of divisions within front or back matter.

Module tei — *1. The TEI Infrastructure*

Used by back front

Members model.frontPart.drama [castList epilogue performance prologue set] divGen titlePage

---

model.frontPart.drama groups elements which appear at the level of divisions within front or back matter of performance texts only.

Module tei — *1. The TEI Infrastructure*

Used by model.frontPart

Members castList epilogue performance prologue set

---

model.gLike groups elements used to represent individual non-Unicode characters or glyphs.

Module tei — *1. The TEI Infrastructure*

Used by bibl byline castItem closer date dictScrap docImprint entryFree etym form gramGrp interp lem m measureGrp oVar opener pVar rdg re sense series time u w xr macro.paraContent macro.phraseSeq macro.specialPara macro.xtext

Members g

---

model.global groups elements which may appear at any point within a TEI text.

Module tei — *1. The TEI Infrastructure*

Used by address app argument back bibl body byline castGroup castItem castList change cit closer date dictScrap div div1 div2 div3 div4 div5 div6 div7 docImprint docTitle entry entryFree epigraph epilogue etym figure floatingText form front gramGrp graph group hom imprint lem lg list m msItem opener performance person postscript prologue rdg re sense series set sp table text time

titlePage u w xr macro.paraContent macro.phraseSeq macro.phraseSeq.limited macro.specialPara

Members model.global.edit [addSpan damageSpan delSpan gap space] model.global.meta [alt altGrp certainty fLib fs fvLib index interp interpGrp join joinGrp link linkGrp respons span spanGrp timeline] model.global.spoken [incident kinesic pause shift vocal writing] model.milestoneLike [anchor cb fw lb milestone pb] model.noteLike [note witDetail]

---

model.global.edit groups globally available elements which perform a specifically editorial function.

Module tei — *1. The TEI Infrastructure*

Used by model.global

Members addSpan damageSpan delSpan gap space

---

model.global.meta groups globally available elements which describe the status of other elements.

Module tei — *1. The TEI Infrastructure*

Used by model.global

Members alt altGrp certainty fLib fs fvLib index interp interpGrp join joinGrp link linkGrp respons span spanGrp timeline

Note Elements in this class are typically used to hold groups of links or of abstract interpretations, or by provide indications of certainty etc. It may find be convenient to localize all metadata elements, for example to contain them within the same divison as the elements that they relate to; or to locate them all to a division of their own. They may however appear at any point in a TEI text.

---

model.global.spoken groups elements which may appear globally within spoken texts.

Module spoken — *8. Transcriptions of Speech*

Used by model.global

Members incident kinesic pause shift vocal writing

Note This class groups elements which can appear anywhere within transcribed speech.

---

model.glossLike groups elements which provide an alternative name, explanation, or description for any markup construct.

Module tei — *1. The TEI Infrastructure*

Used by attDef category certainty char classSpec elementSpec gap glyph incident interp interpGrp join joinGrp kinesic macroSpec moduleSpec respons schemaSpec surface taxonomy valItem vocal zone

Members altIdent desc equiv gloss

---

model.gramPart groups elements allowed within a <gramGrp> element in a dictionary.

Module dictionaries — *9. Dictionaries*

Used by gramGrp model.formPart

Members model.morphLike [case gen gram iType mood number per tns] colloc gramGrp lbl pos subc usg

---

model.graphicLike groups elements containing images, formulae, and similar objects.

Module tei — *1. The TEI Infrastructure*

Used by char facsimile figure formula glyph surface zone model.phrase

Members binaryObject formula graphic

---

model.headLike groups elements used to provide a title or heading at the start of a text division.

Module tei — *1. The TEI Infrastructure*

Used by argument castGroup climate divGen event figure listBibl listNym listOrg listPerson listPlace listWit msDesc msPart org place population set state table terrain trait model.divTopPart

Members head

---

model.headerPart groups high level elements which may appear more than once in a TEI Header.

Module header — *2. The TEI Header*

Used by teiHeader

Members encodingDesc profileDesc

---

model.hiLike groups phrase-level elements which are typographically distinct but to which no specific function can be attributed.

Module tei — *1. The TEI Infrastructure*

Used by model.highlighted

Members hi

---

model.highlighted groups phrase-level elements which are typographically distinct.

Module tei — *1. The TEI Infrastructure*

Used by bibl model.phrase

Members model.emphLike [code distinct emph foreign gloss ident mentioned soCalled term title] model.hiLike [hi]

---

model.imprintPart groups the bibliographic elements which occur inside imprints.

Module tei — *1. The TEI Infrastructure*

Used by imprint model.biblPart

Members biblScope distributor pubPlace publisher

model.inter groups elements which can appear either within or between paragraph-like elements.

Module tei — *1. The TEI Infrastructure*

Used by change dictScrap entryFree etym form gramGrp lem rdg xr macro.limitedContent macro.paraContent macro.specialPara model.common

Members model.biblLike [bibl biblFull biblStruct msDesc] model.egLike [eg egXML] model.labelLike [desc label] model.listLike [list listBibl listNym listOrg listPerson listPlace listWit] model.oddDecl [classSpec elementSpec listRef macroSpec moduleSpec specGrp] model.oddRef [moduleRef specGrpRef] model.qLike [model.quoteLike [cit quote] q said] model.stageLike [camera caption move sound stage tech view] castList figure table

---

model.lLike groups elements representing metrical components such as verse lines.

Module tei — *1. The TEI Infrastructure*

Used by lg sp model.divPart

Members l

---

model.lPart groups phrase-level elements which may appear within verse only.

Module tei — *1. The TEI Infrastructure*

Used by model.phrase

Members caesura rhyme

---

model.labelLike groups elements used to gloss or explain other parts of a document.

Module tei — *1. The TEI Infrastructure*

Used by application climate event location org place population state terrain trait model.inter

Members desc label

---

model.limitedPhrase groups phrase-level elements excluding those elements primarily intended for transcription of existing sources.

Module tei — *1. The TEI Infrastructure*

Used by catDesc change macro.limitedContent macro.phraseSeq.limited

Members model.emphLike [code distinct emph foreign gloss ident mentioned soCalled term title] model.pPart.data [model.addressLike [address affiliation email] model.dateLike [date time] model.measureLike [depth geo height measure measureGrp num width] model.nameLike [model.nameLike.agent [name orgName persName] model.offsetLike [geogFeat offset] model.persNamePart [addName forename genName nameLink roleName surname] model.placeStateLike [model.placeNamePart [**bloc country district geogName placeName region settlement**] state] lang rs]] model.pPart.editorial [abbr am choice ex expan subst] model.pPart.msdesc [catchwords dimensions handShift heraldry locus material origDate

origPlace secFol signatures stamp watermark] model.phrase.xml [att gi tag val] model.ptrLike [ptr ref]

---

model.listLike groups list-like elements.

Module tei — *1. The TEI Infrastructure*

Used by sourceDesc model.inter

Members list listBibl listNym listOrg listPerson listPlace listWit

---

model.measureLike groups elements which denote a number, a quantity, a measurement, or similar piece of text that conveys some numerical meaning.

Module tei — *1. The TEI Infrastructure*

Used by location measureGrp model.pPart.data

Members depth geo height measure measureGrp num width

---

model.milestoneLike groups milestone-style elements used to represent reference systems.

Module tei — *1. The TEI Infrastructure*

Used by model.global

Members anchor cb fw lb milestone pb

---

model.morphLike groups elements which provide morphological information within a dictionary entry.

Module dictionaries — *9. Dictionaries*

Used by etym model.gramPart

Members case gen gram iType mood number per tns

---

model.msItemPart groups elements which can appear within a manuscript item description.

Module tei — *1. The TEI Infrastructure*

Used by msItem

Members model.quoteLike [cit quote] model.respLike [author editor respStmt] bibl colophon decoNote explicit filiation finalRubric incipit listBibl msItem msItemStruct rubric textLang title

---

model.nameLike groups elements which name or refer to a person, place, or organization.

Module tei — *1. The TEI Infrastructure*

Used by org model.addrPart model.pPart.data

Members model.nameLike.agent [name orgName persName] model.offsetLike [geogFeat offset] model.persNamePart [addName forename genName nameLink roleName surname]

model.placeStateLike [model.placeNamePart [bloc country district geogName placeName region settlement] state] lang rs

Note A superset of the naming elements that may appear in datelines, addresses, statements of responsibility, etc.

---

model.nameLike.agent groups elements which contain names of individuals or corporate bodies.

Module tei — *1. The TEI Infrastructure*

Used by respStmt setting model.nameLike

Members name orgName persName

Note This class is used in the content model of elements which reference names of people or organizations.

---

model.noteLike groups globally-available note-like elements.

Module tei — *1. The TEI Infrastructure*

Used by adminInfo biblStruct char climate event glyph location metDecl monogr msItemStruct notesStmt org place population state terrain trait model.global

Members note witDetail

---

model.oddDecl groups elements which generate declarations in some markup language in ODD documents.

Module tei — *1. The TEI Infrastructure*

Used by schemaSpec specGrp model.inter

Members classSpec elementSpec listRef macroSpec moduleSpec specGrp

---

model.oddRef groups elements which reference declarations in some markup language in ODD documents.

Module tei — *1. The TEI Infrastructure*

Used by specGrp model.inter

Members moduleRef specGrpRef

---

model.offsetLike groups elements which can appear only as part of a place name.

Module tei — *1. The TEI Infrastructure*

Used by location model.nameLike

Members geogFeat offset

---

model.orgStateLike groups elements describing changeable characteristics of an organization which have a definite duration.

Module tei — *1. The TEI Infrastructure*

Used by

Members state

---

model.pLike groups paragraph-like elements.

Module tei — *1. The TEI Infrastructure*

Used by application availability binding bindingDesc broadcast cRefPattern climate correction custodialHist decoDesc editionStmt editorialDecl encodingDesc equipment event exemplum figure handDesc history hyphenation interpretation langKnowledge layoutDesc metDecl msContents msDesc msItem msItemStruct msPart normalization nym objectDesc org particDesc person personGrp physDesc place population projectDesc publicationStmt quotation recordHist recording recordingStmt refsDecl relationGrp remarks samplingDecl scriptStmt seal sealDesc segmentation seriesStmt setting settingDesc sourceDesc sp state stdVals supportDesc terrain trait model.divPart

Members ab p

---

model.pLike.front groups paragraph-like elements which can occur as direct constituents of front matter.

Module tei — *1. The TEI Infrastructure*

Used by back front

Members argument byline docAuthor docDate docEdition docImprint docTitle epigraph head titlePart

---

model.pPart.data groups phrase-level elements containing names, dates, numbers, measures, and similar data.

Module tei — *1. The TEI Infrastructure*

Used by bibl model.phrase model.limitedPhrase

Members model.addressLike [address affiliation email] model.dateLike [date time] model.measureLike [depth geo height measure measureGrp num width] model.nameLike [model.nameLike.agent [name orgName persName] model.offsetLike [geogFeat offset] model.persNamePart [addName forename genName nameLink roleName surname] model.placeStateLike [model.placeNamePart [bloc country district geogName placeName region settlement] state] lang rs]

---

model.pPart.edit groups phrase-level elements for simple editorial correction and transcription.

Module tei — *1. The TEI Infrastructure*

Used by bibl model.phrase

Members model.pPart.editorial [abbr am choice ex expan subst] model.pPart.transcriptional [add app corr damage del orig reg restore sic supplied unclear]

---

model.pPart.editorial groups phrase-level elements for simple editorial interventions that may be useful both in transcribing and in authoring.

Module tei — *1. The TEI Infrastructure*

Used by model.pPart.edit model.limitedPhrase

Members abbr am choice ex expan subst

---

model.pPart.msdesc groups phrase-level elements used in manuscript description.

Module tei — *1. The TEI Infrastructure*

Used by model.phrase model.limitedPhrase

Members catchwords dimensions handShift heraldry locus material origDate origPlace secFol signatures stamp watermark

---

model.pPart.transcriptional groups phrase-level elements used for editorial transcription of pre-existing source materials.

Module tei — *1. The TEI Infrastructure*

Used by subst model.pPart.edit

Members add app corr damage del orig reg restore sic supplied unclear

---

model.persEventLike groups elements describing specific events in a person's history, for example birth, marriage, or appointment.

Module tei — *1. The TEI Infrastructure*

Used by model.personPart

Members birth death event

Note These are not characteristics of an individual, but often cause an individual to gain such characteristics, or to enter a new state.

---

model.persNamePart groups elements which form part of a personal name.

Module namesdates — *13. Names, Dates, People, and Places*

Used by model.nameLike

Members addName forename genName nameLink roleName surname

---

model.persStateLike groups elements describing changeable characteristics of a person which have a definite duration, for example occupation, residence, or name.

Module tei — *1. The TEI Infrastructure*

Used by model.personPart

Members affiliation education floruit occupation persName residence state

Note These characteristics of an individual are typically a consequence of their own action or that of others.

model.persTraitLike groups elements describing generally unchanging physical or socially-constructed characteristics of a person, for example hair-colour, ethnicity, or sex.

Module tei — *1. The TEI Infrastructure*

Used by model.personPart

Members age faith langKnowledge nationality sex socecStatus trait

Note These characteristics of an individual are typically independent of their volition or action.

model.personLike groups elements which provide information about people and their relationships.

Module tei — *1. The TEI Infrastructure*

Used by listPerson org particDesc

Members org person personGrp

model.personPart groups elements which form part of the description of a person.

Module tei — *1. The TEI Infrastructure*

Used by person personGrp

Members model.persEventLike [birth death event] model.persStateLike [affiliation education floruit occupation persName residence state] model.persTraitLike [age faith langKnowledge nationality sex socecStatus trait] bibl

model.phrase groups elements which can occur at the level of individual words or phrases.

Module tei — *1. The TEI Infrastructure*

Used by byline castItem closer date dictScrap docImprint entryFree etym form gramGrp lem opener rdg re sense time u xr macro.paraContent macro.phraseSeq macro.specialPara

Members model.graphicLike [binaryObject formula graphic] model.highlighted [model.emphLike [code distinct emph foreign gloss ident mentioned soCalled term title] model.hiLike [hi]] model.lPart [caesura rhyme] model.pPart.data [model.addressLike [address affiliation email] model.dateLike [date time] model.measureLike [depth geo height measure measureGrp num width] model.nameLike [model.nameLike.agent [name orgName persName] model.offsetLike [geogFeat offset] model.persNamePart [addName forename genName nameLink roleName surname] model.placeStateLike [model.placeNamePart **[bloc country district geogName placeName region settlement]** state] lang rs]] model.pPart.edit [model.pPart.editorial [abbr am choice ex expan subst] model.pPart.transcriptional [add app corr damage del orig reg restore sic supplied unclear]] model.pPart.msdesc [catchwords dimensions handShift heraldry locus material origDate origPlace secFol signatures stamp watermark] model.phrase.xml [att gi tag val] model.ptrLike [ptr ref] model.ptrLike.form [oRef oVar pRef pVar] model.segLike [c cl m phr s seg w] model.specDescLike [specDesc specList]

Note This class of elements can occur only within larger elements of the class inter or chunk. In prose, this means these elements can occur within paragraphs, list items, lines of verse, etc.

model.phrase.xml groups phrase-level elements used to encode XML constructs such as element names, attribute names, and attribute values

Module tei — *1. The TEI Infrastructure*

Used by model.phrase model.limitedPhrase

Members att gi tag val

---

model.physDescPart groups specialised elements forming part of the physical description of a manuscript or similar written source.

Module tei — *1. The TEI Infrastructure*

Used by

Members accMat additions bindingDesc decoDesc handDesc musicNotation objectDesc sealDesc

---

model.placeEventLike groups elements which describe events at or affecting a place.

Module tei — *1. The TEI Infrastructure*

Used by place

Members event

---

model.placeLike groups elements used to provide information about places and their relationships.

Module tei — *1. The TEI Infrastructure*

Used by listPlace org place

Members place

---

model.placeNamePart groups elements which form part of a place name.

Module tei — *1. The TEI Infrastructure*

Used by location model.placeStateLike

Members bloc country district geogName placeName region settlement

---

model.placeStateLike groups elements which describe changing states of a place.

Module tei — *1. The TEI Infrastructure*

Used by place model.nameLike

Members model.placeNamePart [bloc country district geogName placeName region settlement] state

---

model.placeTraitLike groups elements which describe unchanging traits of a place.

Module tei — *1. The TEI Infrastructure*

Used by place

Members climate location population terrain trait

model.profileDescPart groups elements which may be used inside <profileDesc> and appear multiple times.

Module header — *2. The TEI Header*

Used by profileDesc

Members handNotes langUsage particDesc settingDesc textClass textDesc

---

model.ptrLike groups elements used for purposes of location and reference.

Module tei — *1. The TEI Infrastructure*

Used by application bibl cit eLeaf eTree relatedItem model.phrase model.limitedPhrase

Members ptr ref

---

model.ptrLike.form groups elements used for purposes of location of particular orthographic or pronunciation forms within a dictionary entry.

Module dictionaries — *9. Dictionaries*

Used by model.phrase

Members oRef oVar pRef pVar

---

model.publicationStmtPart groups elements which may appear within the <publicationStmt> element of the TEI Header.

Module tei — *1. The TEI Infrastructure*

Used by publicationStmt

Members address authority availability date distributor idno pubPlace publisher

---

model.qLike groups elements related to highlighting which can appear either within or between chunk-level elements.

Module tei — *1. The TEI Infrastructure*

Used by cit sp model.inter

Members model.quoteLike [cit quote] q said

---

model.quoteLike groups elements used to directly contain quotations.

Module tei — *1. The TEI Infrastructure*

Used by model.qLike model.msItemPart

Members cit quote

---

model.rdgLike groups elements which contain a single reading, other than the lemma, within a textual variation.

Module textcrit — *12. Critical Apparatus*

Used by app rdgGrp

Members rdg

Note This class allows for variants of the <rdg> element to be easily created via TEI customizations.

---

model.rdgPart groups elements which mark the beginning or ending of a fragmentary manuscript or other witness.

Module textcrit — *12. Critical Apparatus*

Used by lem rdg

Members lacunaEnd lacunaStart wit witEnd witStart

Note These elements may appear anywhere within the elements <lem> and <rdg>, and also within any of their constituent elements.

---

model.recordingPart groups elements used to describe details of an audio or video recording.

Module spoken — *8. Transcriptions of Speech*

Used by recording

Members model.dateLike [date time] broadcast equipment respStmt

---

model.resourceLike groups non-textual elements which may appear together with a header and a text to constitute a TEI document.

Module tei — *1. The TEI Infrastructure*

Used by TEI

Members facsimile fsdDecl

---

model.respLike groups elements which are used to indicate intellectual responsibility, for example within a bibliographic element.

Module tei — *1. The TEI Infrastructure*

Used by model.biblPart model.msItemPart

Members author editor respStmt

---

model.segLike groups elements used for arbitrary segmentation.

Module tei — *1. The TEI Infrastructure*

Used by bibl m w model.phrase

Members c cl m phr s seg w

Note The principles on which segmentation is carried out, and any special codes or attribute values

used, should be defined explicitly in the <segmentation> element of the <encodingDesc> within the associated TEI header.

---

model.settingPart groups elements used to describe the setting of a linguistic interaction.

Module tei — *1. The TEI Infrastructure*

Used by setting

Members activity locale

---

model.sourceDescPart groups elements which may be used inside <sourceDesc> and appear multiple times.

Module header — *2. The TEI Header*

Used by sourceDesc

Members recordingStmt scriptStmt

---

model.specDescLike groups elements for referring to specification elements.

Module tei — *1. The TEI Infrastructure*

Used by model.phrase

Members specDesc specList

---

model.stageLike groups elements containing stage directions or similar things defined by the module for performance texts.

Module tei — *1. The TEI Infrastructure*

Used by sp model.inter

Members camera caption move sound stage tech view

Note Stage directions are members of class inter: that is, they can appear between or within component-level elements.

---

model.textDescPart groups elements used to categorise a text for example in terms of its situational parameters.

Module tei — *1. The TEI Infrastructure*

Used by

Members channel constitution derivation domain factuality interaction preparedness

---

model.titlepagePart groups elements which can occur as direct constituents of a title page, such as <docTitle>, <docAuthor>, <docImprint>, or <epigraph>.

Module tei — *1. The TEI Infrastructure*

Used by titlePage

Members binaryObject byline docAuthor docDate docEdition docImprint docTitle epigraph figure graphic imprimatur titlePart

Appendix B

Attribute Classes

---

att.ascribed provides attributes for elements representing speech or action that can be ascribed to a specific individual.

Module tei — *1. The TEI Infrastructure*

Members change incident kinesic move pause q said setting shift sp u vocal writing

Attributes In addition to global attributes

@who indicates the person, or group of people, to whom the element content is ascribed.

Status Optional

Datatype 1–∞ occurrences of `data.pointer` separated by whitespace

Values For transcribed speech, this will typically identify a participant or participant group; in other contexts, it will point to any identified <person> element.

---

att.coordinated elements which can be positioned within a two dimensional coordinate system.

Module transcr — *11. Representation of Primary Sources*

Members surface zone

Attributes In addition to global attributes

@ulx gives the x coordinate value for the upper left corner of a rectangular space.

Status Optional

Datatype `data.numeric`

@uly gives the y coordinate value for the upper left corner of a rectangular space.

Status Optional

Datatype `data.numeric`

@lrx gives the x coordinate value for the lower right corner of a rectangular space.

Status Optional

Datatype `data.numeric`

@lry gives the y coordinate value for the lower right corner of a rectangular space.

Status Optional

Datatype `data.numeric`

att.damaged provides attributes describing the nature of any physical damage affecting a reading.

Module tei — *1. The TEI Infrastructure*

Members damage damageSpan

Attributes In addition to global attributes

@extent indicates approximately how much text is in the damaged area, in letters, minims, inches, or any appropriate unit, where this cannot be deduced from the contents of the tag.

Status Optional

Datatype 1–∞ occurrences of `data.word` separated by whitespace

Values any measurement phrase, e.g. *25 letters*, *2 × 3 inches*.

@hand In the case of damage (deliberate defacement, inking out, etc.) assignable to a distinct hand, signifies the hand responsible for the damage.

Status Optional

Datatype `data.pointer`

Values must be one of the hand identifiers declared in the document header (see section *11.4.1. Document Hands*).

@agent categorizes the cause of the damage, if it can be identified.

Status Optional

Datatype `data.enumerated`

Sample values include: **rubbing** damage results from rubbing of the leaf edges

mildew damage results from mildew on the leaf surface

smoke damage results from smoke

@degree Signifies the degree of damage according to a convenient scale. The <damage> tag with the **degree** attribute should only be used where the text may be read with some confidence; text supplied from other sources should be tagged as <supplied>.

Status Optional

Datatype

```
data.probability | data.certainty
```

Values an alphanumeric categorization of the degree of damage, as *0.4*.

Note The <damage> tag with the **degree** attribute should only be used where the text may be read with confidence despite the damage. It is appropriate where it is desired to record the fact of damage, though this has not affected the readability of the text (as may be the case with weathered inscriptional materials). Where the damage has rendered the text more or less illegible either the <unclear> tag (for partial illegibility) or the <gap> tag (for complete illegibility, with no text supplied) should be used, with the information concerning the damage given in the attribute values of these tags. See section *11.5.2. Use of the <gap>, <del>, <damage>, <unclear>, and <supplied> Elements in Combination* for discussion of the use of these tags in particular circumstances.

@group assigns an arbitrary number to each stretch of damage regarded as forming part of the same physical phenomenon.

Status Mandatory when applicable

Datatype `data.count`

---

att.datable provides attributes for normalization of elements that contain dates, times, or datable events.

Module tei — *1. The TEI Infrastructure*

Members acquisition affiliation age application binding birth bloc climate country custEvent date death district education event faith floruit geogFeat langKnowledge langKnown location nationality occupation orgName origDate origin persName placeName population provenance region relation residence seal settlement sex socecStatus stamp state terrain time trait

Attributes att.datable.w3c (*@period*, *@when*, *@notBefore*, *@notAfter*, *@from*, *@to*) att.datable.iso (*@when-iso*, *@notBefore-iso*, *@notAfter-iso*, *@from-iso*, *@to-iso*)

Note This 'superclass' provides attributes that can be used to provide normalized values of temporal information. By default, the attributes from the att.datable.w3c class are provided. If the module for names & dates is loaded, this class also provides attributes from the att.datable.iso class. In general, the possible values of attributes restricted to the W3C datatypes form a subset of those values available via the ISO 8601 standard. However, the greater expressiveness of the ISO datatypes may not be needed, and there exists much greater software support for the W3C datatypes.

---

att.datable.iso provides attributes for normalization of elements that contain datable events using the ISO 8601 standard.

Module namesdates — *13. Names, Dates, People, and Places*

Members att.datable [acquisition affiliation age application binding birth bloc climate country custEvent date death district education event faith floruit geogFeat langKnowledge langKnown location nationality occupation orgName origDate origin persName placeName population provenance region relation residence seal settlement sex socecStatus stamp state terrain time trait]

Attributes In addition to global attributes

@when-iso supplies the value of a date or time in a standard form.

Status Optional

Datatype `data.temporal.iso`

Values Any string representing a valid date, time, or one of a variety of combinations.

The following are examples of ISO date, time, and date & time formats that are *not* valid W3C format normalizations.

```
<date
   when-iso="1996-09-24T07:25+00">Sept. 24th, 1996 at 3:25 in the
morning</date>
<date
   when-iso="1996-09-24T03:25-04">Sept. 24th, 1996 at 3:25 in the
morning</date>
<time
   when-iso="1999-01-04T20:42-05">4 Jan 1999 at 8:42 pm</time>
<time
   when-iso="1999-W01-1T20,70-05">4 Jan 1999 at 8:42 pm</time>
```

```
<date
   when-iso="2006-05-18T10:03">a few minutes after
ten in the morning on Thu 18 May</date>
<time
   when-iso="03:00">3 A.M.</time>
<time
   when-iso="14">around two</time>
<time
   when-iso="15,5">half past three</time>
```

All of the examples of the **when** attribute in the att.datable.w3c class are also valid with respect to this attribute.

```
He likes to be
punctual. I said <q>
  <time
     when-iso="12">around noon</time>
</q>,
and he showed up at <time
   when-iso="12:00:00">12 O'clock</time>
on the dot.
```

The second occurence of <time> could have been encoded with the **when** attribute, as **12:00:00** is a valid time with respect to the W3C *XML Schema Part 2: Datatypes* specification. The first occurence could not.

Note The value of **when-iso** should be the normalized representation of the date, time, or combined date & time in one of the formats specified in ISO 8601. The **calendar** attribute, if present, describes the calendar of the content of the element, not the calendar used for the **when** or **when-iso** attribute, which is always in the Gregorian or proleptic Gregorian calendar.

@notBefore-iso specifies the earliest possible date for the event in standard form, e.g. yyyy-mm-dd.

Status Optional

Datatype `data.temporal.iso`

Values A normalized form of temporal expression conforming ISO 8601.

@notAfter-iso specifies the latest possible date for the event in standard form, e.g. yyyy-mm-dd.

Status Optional

Datatype `data.temporal.iso`

Values A normalized form of temporal expression conforming ISO 8601.

@from-iso indicates the starting point of the period in standard form.

Status Optional

Datatype `data.temporal.iso`

Values A normalized form of temporal expression conforming ISO 8601.

@to-iso indicates the ending point of the period in standard form.

Status Optional

Datatype `data.temporal.iso`

Values A normalized form of temporal expression conforming ISO 8601.

Note If both **when-iso** and **dur-iso** are specified, the values should be interpreted as indicating a span of time by its starting time (or date) and duration. That is,

```
<date when-iso="2007-06-01" dur-iso="P8D"/>
```

indicates the same time period as

```
<date when-iso="2007-06-01/P8D"/>
```

In providing a 'regularized' form, no claim is made that the form in the source text is incorrect; the regularized form is simply that chosen as the main form for purposes of unifying variant forms under a single heading.

---

att.datable.w3c provides attributes for normalization of elements that contain datable events using the W3C datatypes.

Module tei — *1. The TEI Infrastructure*

Members att.datable [acquisition affiliation age application binding birth bloc climate country custEvent date death district education event faith floruit geogFeat langKnowledge langKnown location nationality occupation orgName origDate origin persName placeName population provenance region relation residence seal settlement sex socecStatus stamp state terrain time trait]

Attributes In addition to global attributes

@period supplies a pointer to some location defining a named period of time within which the datable item is understood to have occurred.

Status Optional

Datatype `data.pointer`

@when supplies the value of a date or time in a standard form, e.g. yyyy-mm-dd.

Status Optional

Datatype `data.temporal.w3c`

Values A normalized form of temporal expression conforming to the W3C *XML Schema Part 2: Datatypes Second Edition*.

Examples of W3C date, time, and date & time formats.

```
<date
   when="1945-10-24">24 Oct 45</date>
<date
   when="1996-09-24T07:25:00Z">September 24th, 1996 at 3:25 in the
morning</date>
<time
   when="1999-01-04T20:42:00-05:00">Jan 4 1999 at 8 pm</time>
<time
   when="14:12:38">fourteen twelve and 38 seconds</time>
<date
   when="1962-10">October of 1962</date>
<date
   when="--06-12">June 12th</date>
<date
   when="---01">the first of the month</date>
<date
   when="--08">August</date>
```

```
<date
   when="2006">MMVI</date>
<date
   when="0056">56 AD</date>
<date
   when="-0056">56 BC</date>
```

```
This list begins in
the year 1632, more precisely on Trinity Sunday, i.e. the Sunday after
Pentecost, in that year the <date
   calendar="Julian"
   when="1632-06-06">27th of May (old style)</date>.
```

```
<opener>
  <dateline>
    <placeName>Dorchester, Village,</placeName>
    <date
       when="1828-03-02">March 2d. 1828.</date>
  </dateline>
  <salute>To
     Mrs. Cornell,</salute> Sunday <time
      when="12:00:00">noon.</time>
</opener>
```

Note The value of the **when** attribute is a string representing a date or time, or both, in any one of the following forms:

- a date in any of the standard formats described by *XML Schema Part 2: Datatypes Second Edition*, i.e. most commonly `yyyy-mm-dd`, but also `yyyy`, `--mm`, `---dd`, `yyyy-mm`, or `--mm-dd`
- a time in the standard format described by *XML Schema Part 2: Datatypes Second Edition*, i.e. `hh:mm:ss`
- a combined date and time in any of the standard formats described by *XML Schema Part 2: Datatypes Second Edition*, i.e. `yyyy-mm-ddThh:mm:ss`

The value of **when** is constrained to be in one of the date, time, or combined date & time formats recognized by the W3C *XML Schema Part 2: Datatypes* specification. This is true whether the date component (if any) is a Gregorian or a proleptic Gregorian date.At the time of this writing, the W3C does not permit the year **0000**, which would normally be used to represent the year 1 BCE. Using the current W3C system, the year 1 BCE is indicated with **-0001**, the year 2 BCE with **-0002**, etc.

> It is the intention of the XML Schema Working Group to allow **0000** as … the lexical representation of 1 BCE (which is a leap year), **-0001** will become the lexical representation of 2 BCE, **-0002** of 3 BCE, etc.

Thus in some cases it may be advantageous to use the **when-iso** attribute to indicate years BCE instead.

@notBefore specifies the earliest possible date for the event in standard form, e.g. yyyy-mm-dd.

Status Optional

Datatype `data.temporal.w3c`

Values A normalized form of temporal expression conforming to the W3C *XML Schema Part 2: Datatypes Second Edition*.

@notAfter specifies the latest possible date for the event in standard form, e.g. yyyy-mm-dd.

Status Optional

Datatype `data.temporal.w3c`

Values A normalized form of temporal expression conforming to the W3C *XML Schema Part 2: Datatypes Second Edition*.

@from indicates the starting point of the period in standard form, e.g. yyyy-mm-dd.

Status Optional

Datatype `data.temporal.w3c`

Values A normalized form of temporal expression conforming to the W3C *XML Schema Part 2: Datatypes Second Edition*.

@to indicates the ending point of the period in standard form, e.g. yyyy-mm-dd.

Status Optional

Datatype `data.temporal.w3c`

Values A normalized form of temporal expression conforming to the W3C *XML Schema Part 2: Datatypes Second Edition*.

---

att.declarable provides attributes for those elements in the TEI Header which may be independently selected by means of the special purpose **decls** attribute.

Module tei — *1. The TEI Infrastructure*

Members availability bibl biblFull biblStruct broadcast correction editorialDecl equipment geoDecl hyphenation interpretation langUsage listBibl listNym listOrg listPerson listPlace metDecl normalization particDesc projectDesc quotation recording refsDecl samplingDecl scriptStmt segmentation settingDesc sourceDesc stdVals textClass textDesc

Attributes In addition to global attributes

@default indicates whether or not this element is selected by default when its parent is selected.

Status Mandatory when applicable

Datatype `data.truthValue`

Legal values are: **true** This element is selected if its parent is selected

false This element can only be selected explicitly, unless it is the only one of its kind, in which case it is selected if its parent is selected. **[Default]**

Note The rules governing the association of declarable elements with individual parts of a TEI text are fully defined in chapter *15.3. Associating Contextual Information with a Text*. Only one element of a particular type may have a **default** attribute with a value of **true**.

---

att.declaring provides attributes for elements which may be independently associated with a particular declarable element within the header, thus overriding the inherited default for that element.

Module tei — *1. The TEI Infrastructure*

Members back body div div1 div2 div3 div4 div5 div6 div7 facsimile floatingText front gloss graphic group ptr ref surface term text u

Attributes In addition to global attributes

@decls identifies one or more declarable elements within the header, which are understood to apply to the element bearing this attribute and its content.

Status Mandatory when applicable

Datatype 1–∞ occurrences of `data.pointer` separated by whitespace

Values must identify a set of declarable elements of different types.

Note The rules governing the association of declarable elements with individual parts of a TEI text are fully defined in chapter *15.3. Associating Contextual Information with a Text*.

---

att.dimensions provides attributes for describing the size of physical objects.

Module tei — *1. The TEI Infrastructure*

Members depth dimensions height space width

Attributes In addition to global attributes

@unit names the unit used for the measurement

Status Optional

Datatype `data.enumerated`

Suggested values include: **cm** (centimetres)

mm (millimetres)

in (inches)

lines lines of text

chars (characters) characters of text

@quantity specifies the length in the units specified

Status Optional

Datatype `data.numeric`

@precision characterizes the precision of the specified measurement.

Status Optional

Datatype `data.certainty`

@scope specifies the applicability of this measurement, where more than one object is being measured.

Status Optional

Datatype `data.enumerated`

Sample values include: **all** measurement applies to all instances.

most measurement applies to most of the instances inspected.

range measurement applies to only the specified range of instances.

---

att.divLike provides attributes common to all elements which behave in the same way as divisions.

Module tei — *1. The TEI Infrastructure*

Members div div1 div2 div3 div4 div5 div6 div7 lg

Attributes att.metrical (*@met*, *@real*, *@rhyme*)

@org (organization) specifies how the content of the division is organized.
Status Optional
Legal values are: **composite** composite content: i.e. no claim is made about the sequence in which the immediate contents of this division are to be processed, or their inter-relationships.
uniform uniform content: i.e. the immediate contents of this element are regarded as forming a logical unit, to be processed in sequence. **[Default]**

@sample indicates whether this division is a sample of the original source and if so, from which part.
Status Optional
Legal values are: **initial** division lacks material present at end in source.
medial division lacks material at start and end.
final division lacks material at start.
unknown position of sampled material within original unknown.
complete division is not a sample. **[Default]**

@part specifies whether or not the division is fragmented by some other structural element, for example a speech which is divided between two or more verse stanzas.
Status Mandatory when applicable
Legal values are: **Y** (yes) the division is incomplete in some respect
N (no) either the division is complete, or no claim is made as to its completeness. **[Default]**
I (initial) the initial part of an incomplete division
M (medial) a medial part of an incomplete division
F (final) the final part of an incomplete division
Note The values **I**, **M**, or **F** should be used only where it is clear how the division is to be reconstituted.

---

att.duration provides attributes for normalization of elements that contain datable events.

Module spoken — *8. Transcriptions of Speech*

Members date gap recording time

Attributes att.duration.w3c (*@dur*) att.duration.iso (*@dur-iso*)

Note This 'superclass' provides attributes that can be used to provide normalized values of temporal information. By default, the attributes from the att.duration.w3c class are provided. If the module for names & dates is loaded, this class also provides attributes from the att.duration.iso class. In general, the possible values of attributes restricted to the W3C datatypes form a subset of those values available via the ISO 8601 standard. However, the greater expressiveness of the ISO datatypes is rarely needed, and there exists much greater software support for the W3C datatypes.

---

att.duration.iso attributes for recording normalized temporal durations.

Module namesdates — *13. Names, Dates, People, and Places*

Members att.duration [date gap recording time]

Attributes In addition to global attributes

@dur-iso (duration) indicates the length of this element in time.
Status Optional
Datatype `data.duration.iso`

Note If both **when** and **dur** or **dur-iso** are specified, the values should be interpreted as indicating a span of time by its starting time (or date) and duration. In order to represent a time range by a duration and its ending time the **when-iso** attribute must be used.In providing a 'regularized' form, no claim is made that the form in the source text is incorrect; the regularized form is simply that chosen as the main form for purposes of unifying variant forms under a single heading.

---

att.duration.w3c attributes for recording normalized temporal durations.

Module tei — *1. The TEI Infrastructure*

Members att.duration [date gap recording time] att.timed [incident kinesic pause u vocal writing]

Attributes In addition to global attributes

@dur (duration) indicates the length of this element in time.
Status Optional
Datatype `data.duration.w3c`

Note If both **when** and **dur** are specified, the values should be interpreted as indicating a span of time by its starting time (or date) and duration. In order to represent a time range by a duration and its ending time the **when-iso** attribute must be used.In providing a 'regularized' form, no claim is made that the form in the source text is incorrect; the regularized form is simply that chosen as the main form for purposes of unifying variant forms under a single heading.

---

att.editLike provides attributes describing the nature of a encoded scholarly intervention or interpretation of any kind.

Module tei — *1. The TEI Infrastructure*

Members att.transcriptional [add addSpan del delSpan restore subst] affiliation age am birth climate corr date death education event ex expan faith floruit gap langKnowledge langKnown location nationality occupation org orgName origDate origPlace origin persName person place placeName population reg relation residence sex socecStatus state supplied terrain time trait unclear

Attributes In addition to global attributes

@cert (certainty) signifies the degree of certainty associated with the intervention or interpretation.
Status Optional
Datatype `data.certainty`

@resp (responsible party) indicates the agency responsible for the intervention or interpretation, for example an editor or transcriber.
Status Optional
Datatype 1–∞ occurrences of `data.pointer` separated by whitespace
Values A pointer to an element in the document header that is associated with a person asserted as responsible for some aspect of the text's creation, transcription, editing, or encoding.

@evidence indicates the nature of the evidence supporting the reliability or accuracy of the intervention or interpretation.
Status Optional
Datatype `data.enumerated`
Suggested values include: **internal** there is internal evidence to support the intervention.
external there is external evidence to support the intervention.
conjecture the intervention or interpretation has been made by the editor, cataloguer, or scholar on the basis of their expertise.

@source contains a list of one or more pointers indicating the sources which support the given reading.
Status Mandatory when applicable
Datatype 1–∞ occurrences of `data.pointer` separated by whitespace
Values A space-delimited series of sigla; each sigil should correspond to a witness or witness group and occur as the value of the **xml:id** attribute on a <witness> or <msDesc> element elsewhere in the document.

@precision characterizes the precision of the specified intervention or dating.
Status Optional
Datatype `data.certainty`

Note The members of this attribute class are typically used to represent any kind of editorial intervention in a text, for example a correction or interpretation, or to date or localize manuscripts etc.

---

att.enjamb (enjambement) groups elements bearing the **enjamb** attribute.

Module verse — *6. Verse*
Members l
Attributes In addition to global attributes

@enjamb (enjambement) indicates that the end of a verse line is marked by enjambement.
Status Optional
Datatype `data.enumerated`
Sample values include: **no** the line is end-stopped
yes the line in question runs on into the next
weak the line is weakly enjambed
strong the line is strongly enjambed
Note The usual practice will be to give the value 'yes' to this attribute when enjambement is being marked, or the values 'weak' and 'strong' if degrees of enjambement are of interest; if no value is given, however, the attribute does not default to a value of 'no'; this allows the attribute to be omitted entirely when enjambement is not of particular interest.

---

att.entryLike groups the different styles of dictionary entries.

Module dictionaries — *9. Dictionaries*

Members entry entryFree superEntry

Attributes In addition to global attributes

@type indicates type of entry, in dictionaries with multiple types.

Status Required when applicable

Datatype `data.enumerated`

Suggested values include: **main** a main entry (default). **[Default]**

hom (homograph) groups information relating to one homograph within an entry.

xref (cross reference) a reduced entry whose only function is to point to another main entry (e.g. for forms of an irregular verb or for variant spellings: *was* pointing to *be*, or *esthete* to *aesthete*).

affix an entry for a prefix, infix, or suffix.

abbr (abbreviation) an entry for an abbreviation.

supplemental a supplemental entry (for use in dictionaries which issue supplements to their main work in which they include updated information about entries).

foreign an entry for a foreign word in a monolingual dictionary.

@sortKey contains a (sortable) character sequence reflecting the entry's alphabetical position in the printed dictionary.

Status Optional

Datatype `data.word`

Values any sequence of characters which, when sorted with the other values, will produced the desired order; specifics of sort key construction are application-dependent.

Note Dictionary order often differs from the collation sequence of machine-readable character sets; in English-language dictionaries, an entry for *4-H* will often appear alphabetized under 'fourh', and *McCoy* may be alphabetized under'maccoy', while *A1*, *A4*, and *A5* may all appear in numeric order 'alphabetized' between 'a-' and 'AA'. The sort key is required if the orthography of the dictionary entry does not suffice to determine its location.

Note The global **n** attribute may be used to encode the homograph numbers attached to entries for homographs.

---

att.global provides attributes common to all elements in the TEI encoding scheme.

Module tei — *1. The TEI Infrastructure*

Members

Attributes att.global.linking (*@corresp*, *@synch*, *@sameAs*, *@copyOf*, *@next*, *@prev*, *@exclude*, *@select*) att.global.analytic (*@ana*) att.global.facs (*@facs*)

@xml:id (identifier) provides a unique identifier for the element bearing the attribute.

Status Optional

Datatype `xsd:ID`

Values any valid XML identifier.

Note The **xml:id** attribute may be used to specify a canonical reference for an element; see section *3.10. Reference Systems*.

@n (number) gives a number (or other label) for an element, which is not necessarily unique within the document.

Status Optional

Datatype 1–∞ occurrences of `data.word` separated by whitespace

Values any string of characters; often, but not necessarily, numeric.

Note The **n** attribute may be used to specify the numbering of chapters, sections, list items, etc.; it may also be used in the specification of a standard reference system for the text.

@xml:lang (language) indicates the language of the element content using a 'tag' generated according to BCP 47

Status Optional

Datatype `data.language`

Values The value must conform to BCP 47. If the value is a private use code (i.e., starts with `x-` or contains `-x-`) it should, and if not it may, match the value of an **ident** attribute of a <language> element supplied in the TEI Header of the current document.

Note If no value is specified for **xml:lang**, the **xml:lang** value for the immediately enclosing element is inherited; for this reason, a value should always be specified on the outermost element (<TEI>).

@rend (rendition) indicates how the element in question was rendered or presented in the source text.

Status Optional

Datatype 1–∞ occurrences of `data.word` separated by whitespace

Values any string of characters; if the typographic rendition of a text is to be systematically recorded, a systematic set of values for the **rend** attribute should be defined.

```
<head
   rend="align(center) case(allcaps)">
  <lb/>To The
<lb/>Duchesse
<lb/>of
<lb/>Newcastle,
<lb/>On Her
<lb/>
  <hi
     rend="case(mixed)">New Blazing-World</hi>.

</head>
```

Note These Guidelines make no binding recommendations for the values of the **rend** attribute; the characteristics of visual presentation vary too much from text to text and the decision to record or ignore individual characteristics varies too much from project to project. Some potentially useful conventions are noted from time to time at appropriate points in the Guidelines.

@rendition points to a description of the rendering or presentation used for this element in the source text.

Status Optional

Datatype 1–∞ occurrences of `data.pointer` separated by whitespace

Values one or more URIs, separated by whitespace.

```
<head
   rendition="#ac #sc">
  <lb/>To The
<lb/>Duchesse
<lb/>of
<lb/>Newcastle,
<lb/>On Her
<lb/>
  <hi
     rendition="#no">New Blazing-World</hi>.

</head>
<!-- elsewhere... -->
<rendition
   xml:id="sc"
   scheme="css">font-variant: smallcaps</rendition>
<rendition
   xml:id="no"
   scheme="css">font-variant: normal</rendition>
<rendition
   xml:id="ac"
   scheme="css">text-align: center</rendition>
```

Note The **rendition** attribute is used in a very similar way to the **class** attribute defined for XHTML but with the important distinction that its function is to describe the appearance of the source text, not necessarily to determine how that text should be presented on screen or paper.Where both **rendition** and **rend** are supplied, the latter is understood to override or complement the former.Each URI provided should indicate a <rendition> element defining the intended rendition in terms of some appropriate style language, as indicated by the **scheme** attribute.

@xml:base provides a base URI reference with which applications can resolve relative URI references into absolute URI references.

Status Optional

Datatype `data.pointer`

Values any syntactically valid URI reference.

```
<div
   type="bibl">
  <head>Bibliography</head>
  <listBibl
     xml:base="http://www.lib.ucdavis.edu/BWRP/Works/">
    <bibl
       n="1">
      <author>
```

```
      <name>Landon, Letitia Elizabeth</name>
     </author>
     <ref
        target="LandLVowOf.sgm">
     <title>The Vow of the Peacock</title>
     </ref>
    </bibl>
    <bibl
       n="2">
     <author>
     <name>Compton, Margaret Clephane</name>
     </author>
     <ref
        target="NortMIrene.sgm">
     <title>Irene, a Poem in Six Cantos</title>
     </ref>
    </bibl>
    <bibl
       n="3">
     <author>
     <name>Taylor, Jane</name>
     </author>
     <ref
        target="TaylJEssay.sgm">
     <title>Essays in Rhyme on Morals and Manners</title>
     </ref>
    </bibl>
   </listBibl>
</div>
```

Note The global attributes described here are made part of the attribute definition list declaration of each element by including a reference to the pattern att.global.attributes in each such declaration.

---

att.global.analytic provides additional global attributes for associating specific analyses or interpretations with appropriate portions of a text.

Module analysis — *17. Simple Analytic Mechanisms*

Members att.global

Attributes In addition to global attributes

@ana (analysis) indicates one or more elements containing interpretations of the element on which the **ana** attribute appears.

Status Optional

Datatype 1–∞ occurrences of `data.pointer` separated by whitespace

Values one or more valid identifiers of one or more interpretive elements (usually <fs> or <interp>), separated by white space.

Note When multiple values are given, they may reflect either multiple divergent interpretations of an ambiguous text, or multiple mutually consistent interpretations of the same passage in different contexts.

---

att.global.facs groups elements corresponding with all or part of an image, because they contain an alternative representation of it, typically but not necessarily a transcription of it.

Module transcr — *11. Representation of Primary Sources*

Members att.global

Attributes In addition to global attributes

@facs (facsimile) points to all or part of an image which corresponds with the content of the element.

Status Optional

Datatype 1–∞ occurrences of `data.pointer` separated by whitespace

Values one or more URIs, separated by whitespace.

---

att.global.linking defines a set of attributes for hypertext and other linking, which are enabled for all elements when the additional tag set for linking is selected.

Module linking — *16. Linking, Segmentation, and Alignment*

Members att.global

Attributes In addition to global attributes

@corresp (corresponds) points to elements that correspond to the current element in some way.

Status Optional

Datatype 1–∞ occurrences of `data.pointer` separated by whitespace

Values one or more URIs, separated by whitespace.

@synch (synchronous) points to elements that are synchronous with the current element.

Status Optional

Datatype 1–∞ occurrences of `data.pointer` separated by whitespace

Values one or more URIs, separated by whitespace.

@sameAs points to an element that is the same as the current element.

Status Optional

Datatype `data.pointer`

Values a URI.

@copyOf points to an element of which the current element is a copy.

Status Optional

Datatype `data.pointer`

Values a URI.

Note Any content of the current element should be ignored. Its true content is that of the element being pointed at.

@next points to the next element of a virtual aggregate of which the current element is part.

Status Optional

Datatype `data.pointer`

Values a URI.

@prev (previous) points to the previous element of a virtual aggregate of which the current element is part.

Status Optional

Datatype `data.pointer`

Values a URI.

@exclude points to elements that are in exclusive alternation with the current element.

Status Optional

Datatype 1–∞ occurrences of `data.pointer` separated by whitespace

Values one or more URIs, separated by whitespace.

@select selects one or more alternants; if one alternant is selected, the ambiguity or uncertainty is marked as resolved. If more than one alternant is selected, the degree of ambiguity or uncertainty is marked as reduced by the number of alternants not selected.

Status Optional

Datatype 1–∞ occurrences of `data.pointer` separated by whitespace

Values one or more URIs, separated by whitespace.

Note This attribute should be placed on an element which is superordinate to all of the alternants from which the selection is being made.

---

att.handFeatures provides attributes describing aspects of the hand in which a manuscript is written.

Module tei — *1. The TEI Infrastructure*

Members handNote handShift

Attributes In addition to global attributes

@scribe gives a standard name or other identifier for the scribe believed to be responsible for this hand.

Status Optional

Datatype `data.name`

Values Any name

@script characterizes the particular script or writing style used by this hand, for example *secretary*, *copperplate*, *Chancery*, *Italian*, etc.

Status Optional

Datatype 1–∞ occurrences of `data.name` separated by whitespace

@medium describes the tint or type of ink, e.g. *brown*, or other writing medium, e.g. *pencil*

Status Optional

Datatype `data.enumerated`

@scope specifies how widely this hand is used in the manuscript.

Status Optional

Legal values are: **sole** only this hand is used throughout the manuscript

major this hand is used through most of the manuscript

minor this hand is used occasionally in the manuscript

---

att.identified provides attributes for elements which can be referenced by means of a **key** attribute.

Module tagdocs — *22. Documentation Elements*

Members attDef classSpec elementSpec macroSpec moduleSpec schemaSpec valItem

Attributes In addition to global attributes

@ident Supplies the identifier by which this element is referenced.
Status **Required**
Datatype `data.name`
Values an XML name

@predeclare Says whether this object should be predeclared in the tei infrastructure module.
Status Optional
Datatype `xsd:boolean`

@module Supplies the name of the module in which this object is to be defined.
Status Optional
Datatype `xsd:NCName`
Values a name of module

@mode specifies the effect of this declaration on its parent module.
Status Optional
Legal values are: **add** this declaration is added to the current definitions **[Default]**
delete this declaration and all of its children are removed from the current setup
change this declaration changes the declaration of the same name in the current definition
replace this declaration replaces the declaration of the same name in the current definition

Note The informal meaning of the values for **mode** is as follows:

add the object should be created (processing any children in **add** mode); raise an error if an object with the same identifier already exists

replace use this object in preference to any existing object with the same identifier, and ignore any children of that object; process any new children in **replace** mode

delete do not process this object or any existing object with the same identifier; raise an error if any new children supplied

change process this object, and process its children, and those of any existing object with the same identifier, in **change** mode

---

att.internetMedia provides attributes for specifying the type of a computer resource using a standard taxonomy.

Module tei — *1. The TEI Infrastructure*

Members binaryObject equiv graphic

Attributes In addition to global attributes

@mimeType (MIME media type) specifies the applicable multimedia internet mail extension (MIME) media type
Status Mandatory when applicable
Datatype `data.word`
Values The value should be a valid MIME media type

Note This attribute class provides attributes for describing a computer resource, typically available over the internet, according to standard taxonomies. At present only a single taxonomy is supported,

the Multipurpose Internet Mail Extensions Media Type system. This system of typology of media types is defined by the Internet Engineering Task Force in RFC 2046. The list of types is maintained by the Internet Assigned Numbers Authority.

---

att.interpLike provides attributes for elements which represent a formal analysis or interpretation.

Module tei — *1. The TEI Infrastructure*

Members interp interpGrp span spanGrp

Attributes In addition to global attributes

@resp (responsible party) indicates who is responsible for the interpretation.

Status Optional

Datatype `data.pointer`

Values A pointer to an element indicating the person responsible for the interpretation, typically to a <respStmt> in the <teiHeader>.

@type indicates what kind of phenomenon is being noted in the passage.

Status Recommended

Datatype `data.enumerated`

Sample values include: **image** identifies an image in the passage.

character identifies a character associated with the passage.

theme identifies a theme in the passage.

allusion identifies an allusion to another text.

@inst (instances) points to instances of the analysis or interpretation represented by the current element.

Status Optional

Datatype 1–∞ occurrences of `data.pointer` separated by whitespace

Values One or more valid identifiers, separated by whitespace.

Note The current element should be an analytic one. The element pointed at should be a textual one.

---

att.lexicographic defines a set of global attributes available on elements in the base tag set for dictionaries.

Module dictionaries — *9. Dictionaries*

Members case colloc def entryFree etym form gen gram gramGrp hom hyph iType lang lbl mood number oRef oVar orth pRef pVar per pos pron re sense subc syll tns usg xr

Attributes In addition to global attributes

@expand gives an expanded form of information presented more concisely in the dictionary

Status Optional

Datatype `text`

Values any string of characters

```
<gramGrp>
```

```
    <pos>n</pos>
  </gramGrp>
```

@norm (normalized) gives a normalized form of information given by the source text in a non-normalized form

Status Optional

Datatype `text`

Values any string of characters

```
  <gramGrp>
    <pos
       norm="noun">n</pos>
  </gramGrp>
```

@split gives the list of split values for a merged form

Status Optional

Datatype `text`

Values any string of characters

@value gives a value which lacks any realization in the printed source text.

Status Optional

Datatype `text`

Values any string of characters

@orig (original) gives the original string or is the empty string when the element does not appear in the source text.

Status Optional

Datatype `text`

Values any string of characters

@location provides a reference to an <anchor> element elsewhere in the document indicating the original location of this component.

Status Optional

Datatype `data.pointer`

Values a valid identifier for an <anchor> element elsewhere in the current document.

@mergedIn gives a reference to another element, where the original appears as a merged form.

Status Optional

Datatype `data.pointer`

Values any valid identifier.

@opt (optional) indicates whether the element is optional or not

Status Optional

Datatype `xsd:boolean`

---

att.measurement provides attributes to represent a regularized or normalized measurement.

Module tei — *1. The TEI Infrastructure*

Members measure measureGrp

Attributes In addition to global attributes

@unit indicates the units used for the measurement, usually using the standard symbol for the desired units.

Status Optional

Datatype `data.enumerated`

Suggested values include: **m** (metre) SI base unit of length

kg (kilogram) SI base unit of mass

s (second) SI base unit of time

Hz (hertz) SI unit of frequency

Pa (pascal) SI unit of pressure or stress

Ω (ohm) SI unit of electric resistance

L (litre) 1 dm^3

t (tonne) 10^3 kg

ha (hectare) 1 hm^2

Å (ångström) 10^{-10} m

mL (millilitre)

cm (centimetre)

dB (decibel) see remarks, below

kbit (kilobit) 10^3 or 1000 bits

Kibit (kibibit) 2^{10} or 1024 bits

kB (kilobyte) 10^3 or 1000 bytes

KiB (kibibyte) 2^{10} or 1024 bytes

MB (megabyte) 10^6 or 1 000 000 bytes

MiB (mebibyte) 2^{20} or 1 048 576 bytes

Note If the measurement being represented is not expressed in a particular unit, but rather is a number of discrete items, the unit **count** should be used, or the **unit** attribute may be left unspecified.Wherever appropriate, a recognised SI unit name should be used (see further `http://www.bipm.org/en/si/`; `http://physics.nist.gov/cuu/Units/`). The list above is indicative rather than exhaustive.

@quantity specifies the number of the specified units that comprise the measurement

Status Optional

Datatype `data.numeric`

@commodity indicates the substance that is being measured

Status Optional

Datatype 1–∞ occurrences of `data.word` separated by whitespace

Note In general, when the commodity is made of discrete entities, the plural form should be used, even when the measurement is of only one of them.

Note This attribute class provides a triplet of attributes that may be used either to regularize the values of the measurement being encoded, or to normalize them with respect to a standard measurement system.

```
<!-- regularization:--><l>So weren't you gonna buy
<measure quantity="0.5" unit="gal" commodity="ice cream">half a gallon</measure>, baby</l>
<!-- normalization: -->
<l>So won't you go and buy <measure quantity="1.893" unit="L" commodity="ice cream">half a
gallon</measure>,
baby?</l>
```

Note The unit should normally be named using the standard abbreviation for an SI unit (see further `http://www.bipm.org/en/si/`; `http://physics.nist.gov/cuu/Units/`). However, encoders may also specify measurements using informally defined units such as **lines** or **characters**.

---

att.metrical defines a set of attributes which certain elements may use to represent metrical information.

Module verse — *6. Verse*

Members att.divLike [div div1 div2 div3 div4 div5 div6 div7 lg] att.segLike [c cl m phr s seg w] l

Attributes In addition to global attributes

@met (metrical structure, conventional) contains a user-specified encoding for the conventional metrical structure of the element.

Status Recommended

Datatype `token`

Values May contain either a standard term for the kind of metrical unit (e.g. *hexameter*) or an encoded representation for the metrical pattern (e.g. +--+-+-+-+-). In either case, the notation used should be documented by a <metDecl> element within the <encodingDesc> of the associated header.

Note Where this attribute is not specified, the metrical pattern for the element concerned is understood to be inherited from its parent.

@real (metrical structure, realized) contains a user-specified encoding for the actual realization of the conventional metrical structure applicable to the element.

Status Required when applicable

Datatype `token`

Values May contain either a standard term for the kind of metrical unit (e.g. *hexameter*) or an encoded representation for the metrical pattern (e.g. +--+-+-+-+-). In either case, the notation used should be documented by a <metDecl> element within the <encodingDesc> of the associated header.

Note Where this attribute is not specified, the metrical realization for the element concerned is understood to be identical to that specified or implied for the **met** attribute.

@rhyme (rhyme scheme) specifies the rhyme scheme applicable to a group of verse lines.

Status Recommended

Datatype `token`

Values By default, the rhyme scheme is expressed as a string of alphabetic characters each corresponding with a rhyming line. Any non-rhyming lines should be represented by a hyphen or an X. Alternative notations may be defined as for **met** by use of the <metDecl> element in the TEI header.

Note When the default notation is used, it does not make sense to specify this attribute on any unit smaller than a line. Nor does the default notation provide any way to record internal rhyme, or to specify non-conventional rhyming practice. These extensions would require user-defined alternative notations.

---

att.msExcerpt (manuscript excerpt) provides attributes used to describe excerpts from a manuscript placed in a description thereof.

Module msdescription — *10. Manuscript Description*

Members explicit incipit msContents msItem msItemStruct quote

Attributes In addition to global attributes

@defective indicates whether the passage being quoted is defective, i.e. incomplete through loss or damage.

Status Optional

Datatype `data.xTruthValue`

Note In the case of an incipit, indicates whether the incipit as given is defective, i.e. the first words of the text as preserved, as opposed to the first words of the work itself. In the case of an explicit, indicates whether the explicit as given is defective, i.e. the final words of the text as preserved, as opposed to what the closing words would have been had the text of the work been whole.

---

att.naming provides attributes common to elements which refer to named persons, places, organizations etc.

Module tei — *1. The TEI Infrastructure*

Members att.personal [addName forename genName nameLink orgName persName roleName surname] affiliation birth bloc climate collection country death district education event geogFeat geogName institution name nationality occupation placeName population pubPlace region relation repository residence rs settlement socecStatus state terrain trait

Attributes In addition to global attributes

@key provides an external means of locating a full definition for the entity (or entities) being named, such as a database record key or other token.

Status Optional

Datatype `data.key`

Values any string of Unicode characters

Note The value may be a unique identifier from a database, or any other externally-defined string identifying the referent.

@ref (reference) provides an explicit means of locating a full definition for the entity being named by means of a URI.

Status Optional

Datatype 1–∞ occurrences of `data.pointer` separated by whitespace

Values any valid URI

Note The value must point directly to one or more XML elements by means of one or more URIs, separated by whitespace. If more than one is supplied, the implication is that the name identifies several distinct entities.

@nymRef (reference to the canonical name) provides a means of locating the canonical form (nym) of the names associated with the object named by the element bearing it.
Status Optional
Datatype 1–∞ occurrences of `data.pointer` separated by whitespace
Values any valid URI
Note The value must point directly to one or more XML elements by means of one or more URIs, separated by whitespace. If more than one is supplied, the implication is that the name is associated with several distinct canonical names.

---

att.personal (attributes for components of personal names) common attributes for those elements which form part of a personal name.
Module tei — *1. The TEI Infrastructure*
Members addName forename genName nameLink orgName persName roleName surname
Attributes att.naming (*@key*, *@ref*, *@nymRef*)

@full indicates whether the name component is given in full, as an abbreviation or simply as an initial.
Status Optional
Legal values are: **yes** the name component is spelled out in full. **[Default]**
abb (abbreviated) the name component is given in an abbreviated form.
init (initial letter) the name component is indicated only by one initial.

@sort specifies the sort order of the name component in relation to others within the personal name.
Status Optional
Datatype `xsd:nonNegativeInteger`
Values A positive number indicating the sort order.

---

att.placement provides attributes for describing where on the source page or object a textual element appears.
Module tei — *1. The TEI Infrastructure*
Members add addSpan figure fw note witDetail
Attributes In addition to global attributes

@place *Status* Recommended
Datatype `data.enumerated`
Suggested values include: **infralinear** below the line
margin-bot (bottom margin) in the bottom margin
margin-left (left margin) in the left margin
margin-right (right margin) in the right margin
margin-top (top margin) in the top margin
opposite on the opposite, i.e. facing, page
overleaf on the other side of the leaf

supralinear above the line

verso on verso of the sheet or page

app (apparatus) note appears in the apparatus at the foot of the page.

end note appears at end of chapter or volume.

foot note appears at foot of page.

inline note appears as a marked paragraph in the body of the text.

interlinear note appears between lines of the text; a less precise form of either **infralinear** or **supralinear**.

inline addition is made in a space left in the witness by an earlier scribe.

mixed one or more of the other values

---

att.pointing defines a set of attributes used by all elements which point to other elements by means of one or more URI references.

Module linking — *16. Linking, Segmentation, and Alignment*

Members att.pointing.group [altGrp joinGrp linkGrp] alt join link ptr ref

Attributes In addition to global attributes

@type categorizes the pointer in some respect, using any convenient set of categories.

Status Optional

Datatype `data.enumerated`

Values The type should indicate the intended function of the pointer, or the rhetorical relationship between its source and target.

@evaluate specifies the intended meaning when the target of a pointer is itself a pointer.

Status Optional

Legal values are: **all** if the element pointed to is itself a pointer, then the target of that pointer will be taken, and so on, until an element is found which is not a pointer.

one if the element pointed to is itself a pointer, then its target (whether a pointer or not) is taken as the target of this pointer.

none no further evaluation of targets is carried out beyond that needed to find the element specified in the pointer's target.

Note If no value is given, the application program is responsible for deciding (possibly on the basis of user input) how far to trace a chain of pointers.

---

att.pointing.group defines a set of attributes common to all elements which enclose groups of pointer elements.

Module linking — *16. Linking, Segmentation, and Alignment*

Members altGrp joinGrp linkGrp

Attributes att.pointing (*@type*, *@evaluate*)

@domains optionally specifies the identifiers of the elements within which all elements indicated by the contents of this element lie.

Status Optional

Datatype 1–∞ occurrences of `data.name` separated by whitespace

Values a list of at least two generic identifier names, separated by whitespace.

Note If this attribute is supplied every element specified as a target must be contained within the element or elements named by it. An application may choose whether or not to report failures to satisfy this constraint as errors, but may not access an element of the right identifier but in the wrong context. If this attribute is not supplied, then target elements may appear anywhere within the target document.

@targFunc (target function) describes the function of each of the values of the **targets** attribute of the enclosed <link>, <join>, or <alt> tags.

Status Optional

Datatype 1–∞ occurrences of `data.word` separated by whitespace

Values a list of at least two valid names, separated by whitespace.

Note The number of separate values must match the number of values in the **targets** attribute in the enclosed <link>, <join>, or <alt> tags (an intermediate <ptr> element may be needed to accomplish this). It should also match the number of values in the **domains** attribute, of the current element, if one has been specified.

---

att.ptrLike.form (form pointers) common attributes for elements in the dictionary base which point at orthographic or pronunciation forms of the headword.

Module dictionaries — *9. Dictionaries*

Members oRef oVar pRef pVar

Attributes In addition to global attributes

@target identifies the orthographic form or pronunciation referred to.

Status Optional

Datatype `data.pointer`

Values a valid identifier, used on some <orth>, <pron> or <form> element elsewhere in the current document.

---

att.rdgPart attributes for elements which mark the beginning or ending of a fragmentary manuscript or other witness.

Module textcrit — *12. Critical Apparatus*

Members lacunaEnd lacunaStart wit witEnd witStart

Attributes In addition to global attributes

@wit (witness or witnesses) contains a list of one or more sigla indicating the witnesses which begin or end at this point.

Status Mandatory when applicable

Datatype 1–∞ occurrences of `data.pointer` separated by whitespace

Values A space-delimited series of sigla; each sigil should correspond to a witness or witness group and occur as the value of the **xml:id** attribute on a <witness> element elsewhere in the document.

Note These elements may appear anywhere within the elements <lem> and <rdg>, and also within any of their constituent elements.

---

att.segLike provides attributes for elements used for arbitrary segmentation.

Module tei — *1. The TEI Infrastructure*

Members c cl m phr s seg w

Attributes att.metrical (*@met*, *@real*, *@rhyme*)

@type characterizes the type of segment.

Status Optional

Datatype `data.enumerated`

Values For a <cl> may take values such as finite, nonfinite, declarative, interrogative, relative etc. For a <phr> or <w>, values such as noun, verb, preposition, etc., may be used. For an <m> element, values such as clitic, prefix, stem will be more appropriate. For a <c> element, values such as letter, punctuation, digit may be used.

@function characterizes the function of the segment.

Status Optional

Datatype `data.enumerated`

Values For a <cl>, may take values such as coordinate, subject, adverbial etc. For a <phr>, such values as subject, predicate etc. may be more appropriate.

@part specifies whether or not the segment is fragmented by some other structural element, for example a clause which is divided between two or more sentences.

Status Mandatory when applicable

Legal values are: **Y** (yes) the segment is incomplete in some respect

N (no) either the segment is complete, or no claim is made as to its completeness **[Default]**

I (initial) the initial part of an incomplete segment

M (medial) a medial part of an incomplete segment

F (final) the final part of an incomplete segment

Note The values **I**, **M**, or **F** should be used only where it is clear how the division is to be reconstituted.

---

att.spanning provides attributes for elements which delimit a span of text by pointing mechanisms rather than by enclosing it.

Module tei — *1. The TEI Infrastructure*

Members addSpan damageSpan delSpan index

Attributes In addition to global attributes

@spanTo indicates the end of a span initiated by the element bearing this attribute.

Status Mandatory when applicable

Datatype `data.pointer`

Values points to an element following this one in the current document.

Note The span is defined as running in document order from the start of the content of the pointing element (if any) to the end of the content of the element pointed to by the spanTo attribute (if any). If no value is supplied for the attribute, the assumption is that the span is coextensive with the pointing element.

---

att.tableDecoration provides attributes used to decorate rows or cells of a table.

Module tei — *1. The TEI Infrastructure*

Members cell row

Attributes In addition to global attributes

@role indicates the kind of information held in this cell or in each cell of this row.

Status Optional

Datatype `data.enumerated`

Suggested values include: **label** labelling or descriptive information only.

data data values. **[Default]**

Note When this attribute is specified on a row, its value is the default for all cells in this row. When specified on a cell, its value overrides any default specified by the **role** attribute of the parent <row> element.

@rows indicates the number of rows occupied by this cell or row.

Status Optional

Datatype `data.count`

Values A number; a value greater than one indicates that this cell (or row) spans several rows.

Note Where several cells span several rows, it may be more convenient to use nested tables.

@cols (columns) indicates the number of columns occupied by this cell or row.

Status Optional

Datatype `data.count`

Values A number; a value greater than one indicates that this cell or row spans several columns.

Note Where an initial cell spans an entire row, it may be better treated as a heading.

---

att.textCritical defines a set of attributes common to all elements representing variant readings in text critical work.

Module textcrit — *12. Critical Apparatus*

Members lem rdg rdgGrp

Attributes In addition to global attributes

@wit (witness or witnesses) contains a list of one or more pointers indicating the witnesses which attest to a given reading.

Status Mandatory when applicable

Datatype 1–∞ occurrences of `data.pointer` separated by whitespace

Values A space-delimited series of sigla; each sigil should correspond to a witness

or witness group and occur as the value of the **xml:id** attribute on a <witness> element elsewhere in the document.

Note If the apparatus contains readings only for a single witness, this attribute may be consistently omitted.This attribute may occur both within an apparatus gathering variant readings in the transcription of an individual witness and within an apparatus gathering readings from different witnesses.Additional descriptions or alternative versions of the sigla referenced may be supplied as the content of a child <wit> element.

@type classifies the reading according to some useful typology.

Status Optional

Datatype `data.enumerated`

Sample values include: **substantive** the reading offers a substantive variant.

orthographic the reading differs only orthographically, not in substance, from other readings.

@cause classifies the cause for the variant reading, according to any appropriate typology of possible origins.

Status Optional

Datatype `data.enumerated`

Sample values include: **homeoteleuton**

homeoarchy

paleographicConfusion

haplography

dittography

falseEmendation

@varSeq (variant sequence) provides a number indicating the position of this reading in a sequence, when there is reason to presume a sequence to the variants on any one lemma.

Status Optional

Datatype `data.count`

Values a positive integer

Note Different variant sequences could be coded with distinct number trails: 1-2-3 for one sequence, 5-6-7 for another. More complex variant sequences, with (for example) multiple branchings from single readings, may be expressed through the <join> element.

@resp (responsible party) identifies the editor responsible for asserting a particular reading in the witness.

Status Optional

Datatype `data.pointer`

Values A pointer to an element in the document header that is associated with a person asserted as responsible for some aspect of the text's creation, transcription, editing, or encoding (see chapter *21. Certainty and Responsibility*).

Note This attribute is only available within an apparatus gathering variant readings

in the transcription of an individual witness. It may not occur in an apparatus gathering readings from different witnesses.

@hand signifies the hand responsible for a particular reading in the witness.

Status Optional

Datatype `data.pointer`

Values must be one of the hand identifiers declared in the document header (see section *11.4.1. Document Hands*).

Note This attribute is only available within an apparatus gathering variant readings in the transcription of an individual witness. It may not occur in an apparatus gathering readings from different witnesses.

Note This element class defines attributes inherited by <rdg>, <lem>, and <rdgGrp>.

---

att.timed provides attributes common to those elements which have a duration in time, expressed either absolutely or by reference to an alignment map.

Module tei — *1. The TEI Infrastructure*

Members incident kinesic pause u vocal writing

Attributes att.duration.w3c (*@dur*)

@start indicates the location within a temporal alignment at which this element begins.

Status Optional

Datatype `data.pointer`

Note If no value is supplied, the element is assumed to follow the immediately preceding element at the same hierarchic level.

@end indicates the location within a temporal alignment at which this element ends.

Status Optional

Datatype `data.pointer`

Note If no value is supplied, the element is assumed to precede the immediately following element at the same hierarchic level.

---

att.transcriptional provides attributes specific to elements encoding authorial or scribal intervention in a text when transcribing manuscript or similar sources.

Module tei — *1. The TEI Infrastructure*

Members add addSpan del delSpan restore subst

Attributes att.editLike (*@cert*, *@resp*, *@evidence*, *@source*, *@precision*)

@hand signifies the hand of the agent which made the intervention.

Status Optional

Datatype `data.pointer`

Values must refer to a <handNote> element, typically declared in the document header (see section *11.4.1. Document Hands*).

@status indicates the effect of the intervention, for example in the case of a deletion, strikeouts which include too much or too little text, or in the case of an addition, an insertion which duplicates some of the text already present.

Status Optional

Datatype `data.enumerated`

Sample values include: **duplicate** all of the text indicated as an addition duplicates some text that is in the original, whether the duplication is word-for-word or less exact.

duplicate-partial part of the text indicated as an addition duplicates some text that is in the original

excessStart some text at the beginning of the deletion is marked as deleted even though it clearly should not be deleted.

excessEnd some text at the end of the deletion is marked as deleted even though it clearly should not be deleted.

shortStart some text at the beginning of the deletion is not marked as deleted even though it clearly should be.

shortEnd some text at the end of the deletion is not marked as deleted even though it clearly should be.

partial some text in the deletion is not marked as deleted even though it clearly should be.

unremarkable the deletion is not faulty. **[Default]**

Note Status information on each deletion is needed rather rarely except in critical editions from authorial manuscripts; status information on additions is even less common.Marking a deletion or addition as faulty is inescapably an interpretive act; the usual test applied in practice is the linguistic acceptability of the text with and without the letters or words in question.

@seq (sequence) assigns a sequence number related to the order in which the encoded features carrying this attribute are believed to have occurred.

Status Mandatory when applicable

Datatype `data.count`

---

att.translatable provides attributes used to indicate the status of a translatable portion of an ODD document.

Module tei — *1. The TEI Infrastructure*

Members desc exemplum gloss remarks

Attributes In addition to global attributes

@version specifies the version name or number of the source from which the translated version was derived

Status Optional

Datatype `data.word`

Note The version may be a number, a letter, or a date

---

att.typed provides attributes which can be used to classify or subclassify elements in any way.

Module tei — *1. The TEI Infrastructure*

Members ab accMat add addName addSpan altIdent altIdentifier anchor application bibl biblStruct bloc camera charProp cit climate colloc corr country custEvent damage damageSpan date

decoNote del delSpan district div div1 div2 div3 div4 div5 div6 div7 eLeaf eTree event exemplum explicit filiation finalRubric floatingText forename g genName geogFeat gloss head ident incident incipit kinesic lg listBibl listNym listOrg listPerson listPlace location mapping measureGrp msName nameLink nym offset org orgName pause persName place placeName population quote re reg region relatedItem relationGrp restore rhyme roleName rubric seal settlement stamp state surname term terrain text time trait vocal writing

Attributes In addition to global attributes

@type characterizes the element in some sense, using any convenient classification scheme or typology.

Status Optional

Datatype `data.enumerated`

@subtype provides a sub-categorization of the element, if needed

Status Optional

Datatype `data.enumerated`

Note The **subtype** attribute may be used to provide any sub-classification for the element, additional to that provided by its **type** attribute.

Note The typology used may be formally defined using the <classification> element of the <encodingDesc> within the associated TEI header, or as a list within one of the components of the <encodingDesc> element, or informally as descriptive prose within the <encodingDesc> element.

---

att.xmlspace groups TEI elements for which it is reasonable to specify whitespace management using the W3C-defined xml:space attribute.

Module tei — *1. The TEI Infrastructure*

Members eg egXML

Attributes In addition to global attributes

@xml:space signals an intention that white space should be preserved by applications

Status Optional

Legal values are: **default**

preserve

Note The XML specification should be consulted for guidance on the use of this attribute.

Appendix C

Elements

<TEI> (TEI document) contains a single TEI-conformant document, comprising a TEI header and a text, either in isolation or as part of a <teiCorpus> element.

Module textstructure — *4. Default Text Structure*

Attributes In addition to global attributes

@version The version of the TEI scheme

Status Optional

Datatype `xsd:decimal`

Values A number identifying the version of the TEI guidelines

Used by teiCorpus

May contain

header: teiHeader

iso-fs: fsdDecl

textstructure: text

transcr: facsimile

Declaration

```
element TEI
{
   att.global.attributes,
   attribute version { xsd:decimal }?,
   ( teiHeader, ( ( model.resourceLike+, text? ) | text ) )
}
```

<sch:ns prefix="tei" uri="http://www.tei-c.org/ns/1.0"/>
<sch:ns prefix="rng" uri="http://relaxng.org/ns/structure/1.0"/>

Example

```
<TEI>
  <teiHeader>
    <fileDesc>
      <titleStmt>
        <title>The shortest TEI Document Imaginable</title>
      </titleStmt>
```

```
      <publicationStmt>
        <p>First published as part of TEI P2.</p>
      </publicationStmt>
      <sourceDesc>
        <p>No source: this is an original work.</p>
      </sourceDesc>
    </fileDesc>
  </teiHeader>
  <text>
    <body>
      <p>This is about the shortest TEI document imaginable.</p>
    </body>
  </text>
</TEI>
```

Note This element is required.

<ab> (anonymous block) contains any arbitrary component-level unit of text, acting as an anonymous container for phrase or inter level elements analogous to, but without the semantic baggage of, a paragraph.

Module linking — *16. Linking, Segmentation, and Alignment*

Attributes att.typed (*@type*, *@subtype*)

@part specifies whether or not the block is complete.

Status Mandatory when applicable

Legal values are: **Y** (yes) the block is incomplete

N (no) either the block is complete, or no claim is made as to its completeness **[Default]**

I (initial) the initial part of an incomplete block

M (medial) a medial part of an incomplete block

F (final) the final part of an incomplete block

Note The values **I**, **M**, or **F** should be used only where it is clear how the block is to be reconstituted.

Used by model.pLike

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address bibl biblStruct binaryObject cb choice cit corr date del desc distinct email emph expan foreign gap gloss graphic hi index label lb list listBibl measure measureGrp mentioned milestone name note num orig pb ptr q quote ref reg rs said sic soCalled stage term time title unclear

dictionaries: lang oRef oVar pRef pVar

drama: camera caption castList move sound tech view

figures: figure formula table

gaiji: g

header: biblFull

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material msDesc origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName listNym listOrg listPerson listPlace nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att classSpec code eg egXML elementSpec gi ident listRef macroSpec moduleRef moduleSpec specDesc specGrp specGrpRef specList tag val

textcrit: app listWit witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element ab
{
   att.global.attributes,
   att.typed.attributes,
   attribute part { "Y" | "N" | "I" | "M" | "F" }?,
   macro.paraContent}
```

Example

```
<div type="book" n="Genesis">
  <div type="chapter" n="1">
    <ab>In the beginning God creaqted the heaven and the earth.</ab>
    <ab>And the earth was without form, and void; and darkness was upon
        the face of the deep. And the spirit of God moved upon the face of the
        waters.</ab>
    <ab>And God said, Let there be light: and there was light.</ab>
<!-- ...-->
  </div>
</div>
```

Note The <ab> element may be used at the encoder's discretion to mark any component-level elements in a text for which no other more specific appropriate markup is defined.

<abbr> (abbreviation) contains an abbreviation of any sort.

Module core — *3. Elements Available in All TEI Documents*

Attributes In addition to global attributes

@type allows the encoder to classify the abbreviation according to some convenient typology.

Status Optional

Datatype `data.enumerated`

Sample values include: **suspension** the abbreviation provides the first letter(s) of the word or phrase, omitting the remainder.

contraction the abbreviation omits some letter(s) in the middle.

brevigraph the abbreviation comprises a special symbol or mark.

superscription the abbreviation includes writing above the line.

acronym the abbreviation comprises the initial letters of the words of a phrase.

title the abbreviation is for a title of address (Dr, Ms, Mr, …)

organization the abbreviation is for the name of an organization.

geographic the abbreviation is for a geographic name.

Note The **type** attribute is provided for the sake of those who wish to classify abbreviations at their point of occurrence; this may be useful in some circumstances, though usually the same abbreviation will have the same type in all occurrences. As the sample values make clear, abbreviations may be classified by the method used to construct them, the method of writing them, or the referent of the term abbreviated; the typology used is up to the encoder and should be carefully planned to meet the needs of the expected use. For a typology of Middle English abbreviations, see Petty (1977)

Used by model.pPart.editorial model.choicePart

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address binaryObject cb choice corr date del distinct email emph expan foreign gap gloss graphic hi index lb measure measureGrp mentioned milestone name note num orig pb ptr ref reg rs sic soCalled term time title unclear

dictionaries: lang oRef oVar pRef pVar

figures: formula

gaiji: g

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident specDesc specList tag val

textcrit: app witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element abbr
{
   att.global.attributes,
   attribute type { data.enumerated }?,
   macro.phraseSeq}
```

Example

```
<abbr>SPQR</abbr>
```

Example

```
<choice>
  <abbr>SPQR</abbr>
  <expan>senatus populusque romanorum</expan>
</choice>
```

Note The <abbr> tag is not required; if appropriate, the encoder may transcribe abbreviations in the source text silently, without tagging them. If abbreviations are not transcribed directly but *expanded* silently, then the TEI header should so indicate.

<accMat> (accompanying material) contains details of any significant additional material which may be closely associated with the manuscript being described, such as non-contemporaneous documents or fragments bound in with the manuscript at some earlier historical period.

Module msdescription — *10. Manuscript Description*

Attributes att.typed (*@type*, *@subtype*)

Used by model.physDescPart

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address bibl biblStruct binaryObject cb choice cit corr date del desc distinct email emph expan foreign gap gloss graphic hi index l label lb lg list listBibl measure measureGrp mentioned milestone name note num orig p pb ptr q quote ref reg rs said sic soCalled sp stage term time title unclear

dictionaries: lang oRef oVar pRef pVar

drama: camera caption castList move sound tech view

figures: figure formula table

gaiji: g

header: biblFull

iso-fs: fLib fs fvLib

linking: ab alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material msDesc origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName listNym listOrg listPerson listPlace nameLink offset orgName persName placeName region roleName settlement state surname

nets: eTree forest forestGrp graph tree

spoken: incident kinesic pause shift u vocal writing

tagdocs: att classSpec code eg egXML elementSpec gi ident listRef macroSpec moduleRef moduleSpec schemaSpec specDesc specGrp specGrpRef specList tag val

textcrit: app listWit witDetail

textstructure: floatingText

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element accMat
{
   att.global.attributes,
   att.typed.attributes,
   macro.specialPara}
```

Example

```
<accMat>A copy of a tax form from 1947 is included in the envelope
with the letter. It is not catalogued separately.</accMat>
```

<acquisition> contains any descriptive or other information concerning the process by which a manuscript or manuscript part entered the holding institution.

Module msdescription — *10. Manuscript Description*

Attributes att.datableatt.datable.w3c (*@period*, *@when*, *@notBefore*, *@notAfter*, *@from*, *@to*) att.datable.iso (*@when-iso*, *@notBefore-iso*, *@notAfter-iso*, *@from-iso*, *@to-iso*)

Used by history

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address bibl biblStruct binaryObject cb choice cit corr date del desc distinct email emph expan foreign gap gloss graphic hi index l label lb lg list listBibl measure measureGrp mentioned milestone name note num orig p pb ptr q quote ref reg rs said sic soCalled sp stage term time title unclear

dictionaries: lang oRef oVar pRef pVar

drama: camera caption castList move sound tech view

figures: figure formula table

gaiji: g

header: biblFull

iso-fs: fLib fs fvLib

linking: ab alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material msDesc origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName listNym listOrg listPerson listPlace nameLink offset orgName persName placeName region roleName settlement state surname

nets: eTree forest forestGrp graph tree

spoken: incident kinesic pause shift u vocal writing

tagdocs: att classSpec code eg egXML elementSpec gi ident listRef macroSpec moduleRef moduleSpec schemaSpec specDesc specGrp specGrpRef specList tag val

textcrit: app listWit witDetail

textstructure: floatingText

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element acquisition
{
   att.global.attributes,
   att.datable.w3c.attributes,
   att.datable.iso.attributes,
   macro.specialPara}
```

Example

```
<acquisition>Left to the <name  type="place">Bodleian</name> by
<name  type="person">Richard Rawlinson</name> in 1755.
</acquisition>
```

<activity> contains a brief informal description of what a participant in a language interaction is doing other than speaking, if anything.

Module corpus — *15. Language Corpora*

Attributes Global attributes only

Used by model.settingPart

May contain

analysis: interp interpGrp span spanGrp

certainty: certainty respons

core: abbr address cb choice date distinct email emph expan foreign gap gloss index lb measure measureGrp mentioned milestone name note num pb ptr ref rs soCalled term time title

dictionaries: lang

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident tag val

textcrit: witDetail

transcr: addSpan am damageSpan delSpan ex fw handShift space subst

Declaration

```
element activity { att.global.attributes, macro.phraseSeq.limited }
```

Example

```
<activity>driving</activity>
```

Note For more fine-grained description of participant activities during a spoken text, the <event> element should be used.

<actor> Name of an actor appearing within a cast list.

Module drama — *7. Performance Texts*

Attributes Global attributes only

Used by model.castItemPart

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address binaryObject cb choice corr date del distinct email emph expan foreign gap gloss graphic hi index lb measure measureGrp mentioned milestone name note num orig pb ptr ref reg rs sic soCalled term time title unclear

dictionaries: lang oRef oVar pRef pVar

figures: formula

gaiji: g

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident specDesc specList tag val

textcrit: app witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element actor { att.global.attributes, macro.phraseSeq }
```

Example

```
<castItem>
  <role>Mathias</role>
  <roleDesc>the Burgomaster</roleDesc>
```

```
  <actor>Mr. Henry Irving</actor>
</castItem>
```

Note This element should be used only to mark the name of the actor as given in the source. Chapter *13. Names, Dates, People, and Places* discusses ways of marking the components of names, and also of associating names with biographical information about a person.

<add> (addition) contains letters, words, or phrases inserted in the text by an author, scribe, annotator, or corrector.

Module core — *3. Elements Available in All TEI Documents*

Attributes att.transcriptional (*@hand*, *@status*, *@seq*) att.editLike (*@cert*, *@resp*, *@evidence*, *@source*, *@precision*) att.placement (*@place*) att.typed (*@type*, *@subtype*)

Used by model.pPart.transcriptional

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address bibl biblStruct binaryObject cb choice cit corr date del desc distinct email emph expan foreign gap gloss graphic hi index label lb list listBibl measure measureGrp mentioned milestone name note num orig pb ptr q quote ref reg rs said sic soCalled stage term time title unclear

dictionaries: lang oRef oVar pRef pVar

drama: camera caption castList move sound tech view

figures: figure formula table

gaiji: g

header: biblFull

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material msDesc origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName listNym listOrg listPerson listPlace nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att classSpec code eg egXML elementSpec gi ident listRef macroSpec moduleRef moduleSpec specDesc specGrp specGrpRef specList tag val

textcrit: app listWit witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element add
{
   att.global.attributes,
```

```
  att.transcriptional.attributes,
  att.editLike.attributes,
  att.placement.attributes,
  att.typed.attributes,
  macro.paraContent}
```

Example

```
The story I am going to relate is true as to
its main facts, and as to the consequences <add  place="supralinear">of
these facts</add> from which this tale takes its title.
```

Note The <add> element should not be used for additions made by editors or encoders. In these cases, either the <corr> or <supplied> element should be used.

<addName> (additional name) contains an additional name component, such as a nickname, epithet, or alias, or any other descriptive phrase used within a personal name.

Module namesdates — *13. Names, Dates, People, and Places*

Attributes att.personal (*@full*, *@sort*) att.naming (*@key*, *@ref*, *@nymRef*) att.typed (*@type*, *@subtype*)

Used by model.persNamePart

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address binaryObject cb choice corr date del distinct email emph expan foreign gap gloss graphic hi index lb measure measureGrp mentioned milestone name note num orig pb ptr ref reg rs sic soCalled term time title unclear

dictionaries: lang oRef oVar pRef pVar

figures: formula

gaiji: g

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident specDesc specList tag val

textcrit: app witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element addName
{
```

```
   att.global.attributes,
   att.personal.attributes,
   att.naming.attributes,
   att.typed.attributes,
   macro.phraseSeq}
```

Example

```
<persName>
  <forename>Frederick</forename>
  <addName  type="epithet">the Great</addName>
  <roleName>Emperor of Prussia</roleName>
</persName>
```

<addSpan/> (added span of text) marks the beginning of a longer sequence of text added by an author, scribe, annotator or corrector (see also «<add>»).

Module transcr — *11. Representation of Primary Sources*

Attributes att.transcriptional (*@hand*, *@status*, *@seq*) att.editLike (*@cert*, *@resp*, *@evidence*, *@source*, *@precision*) att.placement (*@place*) att.typed (*@type*, *@subtype*) att.spanning (*@spanTo*)

Used by model.global.edit

May contain Empty element

Declaration

```
element addSpan
{
   att.global.attributes,
   att.transcriptional.attributes,
   att.editLike.attributes,
   att.placement.attributes,
   att.typed.attributes,
   att.spanning.attributes,
   empty
}
```

<sch:pattern name="spanTo_required"> <sch:rule context="tei:addSpan">
<sch:assert test="@spanTo">The spanTo= attribute of <sch:name/> is required.</sch:assert>
</sch:rule> </sch:pattern>

Example

```
<handNote  xml:id="HEOL"  scribe="HelgiÓlafsson"/>
<!-- ... -->
<body>
  <div>
<!-- text here -->
  </div>
  <addSpan  n="added gathering"  hand="#HEOL"  spanTo="#P025"/>
  <div>
<!-- text of first added poem here -->
  </div>
  <div>
<!-- text of second added poem here -->
```

```
  </div>
  <div>
<!-- text of third added poem here -->
  </div>
  <div>
<!-- text of fourth added poem here -->
  </div>
  <anchor xml:id="P025"/>
  <div>
<!-- more text here -->
  </div>
</body>
```

Note Both the beginning and the end of the added material must be marked; the beginning by the <addSpan> element itself, the end by the **spanTo** attribute.

<additional>

groups additional information, combining bibliographic information about a manuscript, or surrogate copies of it with curatorial or administrative information.

Module msdescription — *10. Manuscript Description*

Attributes Global attributes only

Used by msDesc msPart

May contain

core: listBibl

msdescription: adminInfo surrogates

Declaration

```
element additional
{
   att.global.attributes,
   ( adminInfo?, surrogates?, listBibl? )
}
```

Example

```
<additional>
  <adminInfo>
    <recordHist>
<!-- record history here -->
    </recordHist>
    <custodialHist>
<!-- custodial history here -->
    </custodialHist>
  </adminInfo>
  <surrogates>
<!-- information about surrogates here -->
  </surrogates>
  <listBibl>
<!-- full bibliography here -->
  </listBibl>
</additional>
```

<additions> contains a description of any significant additions found within a manuscript, such as marginalia or other annotations.

Module msdescription — *10. Manuscript Description*

Attributes Global attributes only

Used by model.physDescPart

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address bibl biblStruct binaryObject cb choice cit corr date del desc distinct email emph expan foreign gap gloss graphic hi index l label lb lg list listBibl measure measureGrp mentioned milestone name note num orig p pb ptr q quote ref reg rs said sic soCalled sp stage term time title unclear

dictionaries: lang oRef oVar pRef pVar

drama: camera caption castList move sound tech view

figures: figure formula table

gaiji: g

header: biblFull

iso-fs: fLib fs fvLib

linking: ab alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material msDesc origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName listNym listOrg listPerson listPlace nameLink offset orgName persName placeName region roleName settlement state surname

nets: eTree forest forestGrp graph tree

spoken: incident kinesic pause shift u vocal writing

tagdocs: att classSpec code eg egXML elementSpec gi ident listRef macroSpec moduleRef moduleSpec schemaSpec specDesc specGrp specGrpRef specList tag val

textcrit: app listWit witDetail

textstructure: floatingText

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element additions { att.global.attributes, macro.specialPara }
```

Example

```
<additions>
  <p>There are several marginalia in this manuscript. Some consist of
     single characters and others are figurative. On 8v is to be found a drawing of
     a mans head wearing a hat. At times sentences occurs: On 5v:
  <q  xml:lang="is">Her er skrif andres isslendin</q>,
```

```
        on 19r: <q xml:lang="is">þeim go</q>,
        on 21r: <q xml:lang="is">amen med aund ok munn halla rei knar hofud summu all huad
           batar þad mælgi ok mal</q>,
        On 21v: some runic letters and the sentence <q xml:lang="la">aue maria gracia plena
    dominus</q>.</p>
    </additions>
```

<addrLine> (address line) contains one line of a postal address.

Module core — *3. Elements Available in All TEI Documents*

Attributes Global attributes only

Used by model.addrPart

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address binaryObject cb choice corr date del distinct email emph expan foreign gap gloss graphic hi index lb measure measureGrp mentioned milestone name note num orig pb ptr ref reg rs sic soCalled term time title unclear

dictionaries: lang oRef oVar pRef pVar

figures: formula

gaiji: g

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident specDesc specList tag val

textcrit: app witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element addrLine { att.global.attributes, macro.phraseSeq }
```

Example

```
<address>
  <addrLine>Computing Center, MC 135</addrLine>
  <addrLine>P.O. Box 6998</addrLine>
  <addrLine>Chicago, IL</addrLine>
  <addrLine>60680 USA</addrLine>
</address>
```

Note Addresses may be encoded either as a sequence of lines, or using any sequence of component elements from the model.addrPart class. Other non-postal forms of address, such as telephone numbers or email, should not be included within an <address> element directly but may be wrapped within an <addrLine> if they form part of the printed address in some source text.

<address> contains a postal address, for example of a publisher, an organization, or an individual.

Module core — *3. Elements Available in All TEI Documents*

Attributes Global attributes only

Used by model.addressLike model.publicationStmtPart

May contain

analysis: interp interpGrp span spanGrp

certainty: certainty respons

core: addrLine cb gap index lb milestone name note pb postBox postCode rs street

dictionaries: lang

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp timeline

namesdates: addName bloc country district forename genName geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

textcrit: witDetail

transcr: addSpan damageSpan delSpan fw space

Declaration

```
element address
{
   att.global.attributes,
   ( model.global*, ( ( model.addrPart ), model.global* )+ )
}
```

Example

```
<address>
  <street>via Marsala 24</street>
  <postCode>40126</postCode>
  <name>Bologna</name>
  <name  n="I">Italy</name>
</address>
```

Example

```
<address>
  <addrLine>Computing Center, MC 135</addrLine>
  <addrLine>P.O. Box 6998</addrLine>
  <addrLine>Chicago, IL 60680</addrLine>
  <addrLine>USA</addrLine>
</address>
```

Note This element should be used for postal addresses only. Within it, the generic element <addrLine> may be used as an alternative to any of the more specialized elements available from the model.addrPart class, such as <street>, <postCode> etc.

<adminInfo> (administrative information) contains information about the present custody and availability of the manuscript, and also about the record description itself.

Module msdescription — *10. Manuscript Description*

Attributes Global attributes only

Used by additional

May contain

core: note

header: availability

msdescription: custodialHist recordHist

textcrit: witDetail

Declaration

```
element adminInfo
{
   att.global.attributes,
   ( recordHist?, availability?, custodialHist?, model.noteLike? )
}
```

Example

```
<adminInfo>
  <recordHist>
    <source>Record created <date>1 Aug 2004</date>
    </source>
  </recordHist>
  <availability>
    <p>Until 2015 permission to photocopy some materials from this
        collection has been limited at the request of the donor. Please ask repository staff for
details
        if you are interested in obtaining photocopies from Series 1:
        Correspondence.</p>
  </availability>
  <custodialHist>
    <p>Collection donated to the Manuscript Library by the Estate of
        Edgar Holden in 1993. Donor number: 1993-034.</p>
  </custodialHist>
</adminInfo>
```

<affiliation> contains an informal description of a person's present or past affiliation with some organization, for example an employer or sponsor.

Module namesdates — *13. Names, Dates, People, and Places*

Attributes att.editLike (*@cert*, *@resp*, *@evidence*, *@source*, *@precision*) att.datableatt.datable.w3c (*@period*, *@when*, *@notBefore*, *@notAfter*, *@from*, *@to*) att.datable.iso (*@when-iso*, *@notBefore-iso*, *@notAfter-iso*, *@from-iso*, *@to-iso*) att.naming (*@key*, *@ref*, *@nymRef*)

Used by model.addressLike model.persStateLike

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address binaryObject cb choice corr date del distinct email emph expan foreign gap gloss graphic hi index lb measure measureGrp mentioned milestone name note num orig pb ptr ref reg rs sic soCalled term time title unclear

dictionaries: lang oRef oVar pRef pVar

figures: formula

gaiji: g

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident specDesc specList tag val

textcrit: app witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element affiliation
{
   att.global.attributes,
   att.editLike.attributes,
   att.datable.w3c.attributes,
   att.datable.iso.attributes,
   att.naming.attributes,
   macro.phraseSeq}
```

Example

```
<affiliation>Junior project officer for the US
<name  type="org">National Endowment for the Humanities</name>
</affiliation>
<affiliation  notAfter="1960-01-01"  notBefore="1957-02-28">Paid up
member of the <orgName>Australian Journalists Association</orgName>
</affiliation>
```

Note If included, the name of an organization may be tagged using either the <name> element as above, or the more specific <orgName> element.

<age> specifies the age of a person.

Module namesdates — *13. Names, Dates, People, and Places*

Attributes att.editLike (*@cert*, *@resp*, *@evidence*, *@source*, *@precision*) att.datableatt.datable.w3c (*@period*, *@when*, *@notBefore*, *@notAfter*, *@from*, *@to*) att.datable.iso (*@when-iso*, *@notBefore-iso*, *@notAfter-iso*, *@from-iso*, *@to-iso*)

@value supplies a numeric code representing the age or age group

Status Optional

Datatype `data.count`

Note This attribute may be used to complement a more detailed discussion of a person's age in the content of the element

Used by model.persTraitLike

May contain

analysis: interp interpGrp span spanGrp

certainty: certainty respons

core: abbr address cb choice date distinct email emph expan foreign gap gloss index lb measure measureGrp mentioned milestone name note num pb ptr ref rs soCalled term time title

dictionaries: lang

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident tag val

textcrit: witDetail

transcr: addSpan am damageSpan delSpan ex fw handShift space subst

Declaration

```
element age
{
   att.global.attributes,
   att.editLike.attributes,
   att.datable.w3c.attributes,
   att.datable.iso.attributes,
   attribute value { data.count }?,
   macro.phraseSeq.limited}
```

Example

```
<age value="2" notAfter="1986">under 20 in the early eighties</age>
```

<alt/> (alternation) identifies an alternation or a set of choices among elements or passages.

Module linking — *16. Linking, Segmentation, and Alignment*

Attributes att.pointing (*@type*, *@evaluate*)

@targets specifies the identifiers of the alternative elements or passages.

Status **Required**

Datatype 2–∞ occurrences of `data.pointer` separated by whitespace

Values Each value specified must be the same as that specified as value for an **xml:id** attribute for some other element in the current document.

@mode states whether the alternations gathered in this collection are exclusive or inclusive.

Status Recommended

Legal values are: **excl** (exclusive) indicates that the alternation is exclusive, i.e. that at most one of the alternatives occurs.

incl (inclusive) indicates that the alternation is not exclusive, i.e. that one or more of the alternatives occur.

@weights If **mode** is `excl`, each weight states the probability that the corresponding alternative occurs. If **mode** is **incl** each weight states the probability that the corresponding alternative occurs given that at least one of the other alternatives occurs.

Status Optional

Datatype 2–∞ occurrences of `data.probability` separated by whitespace

Values a whitespace-separated list of probability values in the range from 0 to 1.

Note If **mode** is `excl`, the sum of weights must be 1. If **mode** is `incl`, the sum of weights must be in the range from 0 to the number of alternants.

Used by altGrp model.global.meta

May contain Empty element

Declaration

```
element alt
{
   att.global.attributes,
   att.pointing.attributes,
   attribute targets { list { data.pointer, data.pointer+ } },
   attribute mode { "excl" | "incl" }?,
   attribute weights { list { data.probability, data.probability+ } }?,
   empty
}
```

Example

```
<alt  mode="excl"  targets="#we.fun #we.sun"  weights="0.5 0.5"/>
```

<altGrp> (alternation group) groups a collection of <alt> elements and possibly pointers.

Module linking — *16. Linking, Segmentation, and Alignment*

Attributes att.pointing.group (*@domains*, *@targFunc*) att.pointing (*@type*, *@evaluate*)

@mode states whether the alternations gathered in this collection are exclusive or inclusive.

Status Optional

Legal values are: **excl** (exclusive) indicates that the alternation is exclusive, i.e. that at most one of the alternatives occurs. **[Default]**

incl (inclusive) indicates that the alternation is not exclusive, i.e. that one or more of the alternatives occur.

Used by model.global.meta

May contain

core: ptr

linking: alt

Declaration

```
element altGrp
{
   att.global.attributes,
   att.pointing.group.attributes,
   att.pointing.attributes,
   attribute mode { "excl" | "incl" }?,
   ( alt | ptr )*
}
```

Example

```
<altGrp  mode="excl">
  <alt  targets="#dm #lt #bb"  weights="0.5 0.25 0.25"/>
  <alt  targets="#rl #db"  weights="0.5 0.5"/>
</altGrp>
```

Example

```
<altGrp  mode="incl">
  <alt  targets="#dm #rl"  weights="0.90 0.90"/>
  <alt  targets="#lt #rl"  weights="0.5 0.5"/>
  <alt  targets="#bb #rl"  weights="0.5 0.5"/>
  <alt  targets="#dm #db"  weights="0.10 0.10"/>
  <alt  targets="#lt #db"  weights="0.45 0.90"/>
  <alt  targets="#bb #db"  weights="0.45 0.90"/>
</altGrp>
```

Note Any number of alternations, pointers or extended pointers.

<altIdent>

(alternate identifier) supplies the recommended XML name for an element, class, attribute, etc. in some language.

Module tagdocs — *22. Documentation Elements*

Attributes att.typed (*@type*, *@subtype*)

Used by model.glossLike

May contain

gaiji: g

Declaration

```
element altIdent { att.global.attributes, att.typed.attributes, macro.xtext }
```

Example

```
<altIdent  xml:lang="fr">balisageDoc</altIdent>
```

Note All documentation elements in ODD have a canonical name, supplied as the value for their **ident** attribute. The <altIdent> element is used to supply an alternative name for the corresponding XML object, perhaps in a different language.

<altIdentifier> (alternative identifier) contains an alternative or former structured identifier used for a manuscript, such as a former catalogue number.

Module msdescription — *10. Manuscript Description*

Attributes att.typed (*@type*, *@subtype*)

Used by msIdentifier msPart

May contain

core: note

header: idno

msdescription: collection institution repository

Declaration

```
element altIdentifier
{
   att.global.attributes,
   att.typed.attributes,
   (
      model.placeNamePart_sequenceOptional,
      institution?,
      repository?,
      collection?,
      idno,
      note?
   )
}
```

Example

```
<altIdentifier>
  <settlement>San Marino</settlement>
  <repository>Huntington Library</repository>
  <idno>MS.El.26.C.9</idno>
</altIdentifier>
```

Note An identifying number of some kind must be supplied if known; if it is not known, this should be stated.

<am> (abbreviation marker) contains a sequence of letters or signs present in an abbreviation which are omitted or replaced in the expanded form of the abbreviation.

Module transcr — *11. Representation of Primary Sources*

Attributes att.editLike (*@cert*, *@resp*, *@evidence*, *@source*, *@precision*)

Used by model.pPart.editorial model.choicePart

May contain

gaiji: g

Declaration

```
element am { att.global.attributes, att.editLike.attributes, macro.xtext }
```

Example

```
do you
<abbr>Mr<am>.</am>
</abbr> Jones?
```

<analytic> (analytic level) contains bibliographic elements describing an item (e.g. an article or poem) published within a monograph or journal and not as an independent publication.

Module core — *3. Elements Available in All TEI Documents*

Attributes Global attributes only

Used by biblStruct

May contain

core: author editor respStmt title

Declaration

```
element analytic
{
   att.global.attributes,
   ( author | editor | respStmt | title )*
}
```

Example

```
<biblStruct>
  <analytic>
    <author>Chesnutt, David</author>
    <title>Historical Editions in the States</title>
  </analytic>
  <monogr>
    <title  level="j">Computers and the Humanities</title>
    <imprint>
      <biblScope>25.6</biblScope>
      <date  when="1991-12">(December, 1991):</date>
      <biblScope>377–380</biblScope>
    </imprint>
  </monogr>
</biblStruct>
```

Note May contain titles and statements of responsibility (author, editor, or other), in any order.The <analytic> element may only occur only within a <biblStruct>, where its use is mandatory for the description of an analytic level bibliographic item.

<anchor/> (anchor point) attaches an identifier to a point within a text, whether or not it corresponds with a textual element.

Module linking — *16. Linking, Segmentation, and Alignment*

Attributes att.typed (*@type*, *@subtype*)

Used by model.milestoneLike

May contain Empty element

Declaration

```
element anchor { att.global.attributes, att.typed.attributes, empty }
```

Example

```
<s>The anchor is he<anchor  xml:id="A234"/>re somewhere.</s>
<s>Help me find it.<ptr  target="#A234"/>
</s>
```

Note On this element, the global **xml:id** attribute must be supplied to specify an identifier for the point at which this element occurs within a document. The value used may be chosen freely provided that it is unique within the document and is a syntactically valid name. There is no requirement for values containing numbers to be in sequence.

<app> (apparatus entry) contains one entry in a critical apparatus, with an optional lemma and at least one reading.

Module textcrit — *12. Critical Apparatus*

Attributes In addition to global attributes

@type classifies the variation contained in this element according to some convenient typology.

Status Optional

Datatype `data.enumerated`

Values Any convenient descriptive word or phrase, describing the extent of the variation (e.g. *word*, *phrase*, *punctuation*, etc.) its text-critical significance (e.g. *significant*, *accidental*, *unclear*), or the nature of the variation or the principles required to understand it (e.g. *lectio difficilior*, *usus auctoris*, etc.)

@from identifies the beginning of the lemma in the base text, if necessary.

Status Optional

Datatype `data.pointer`

Values any valid identifier

Note This attribute is only used when the double-end point method of apparatus markup is used.

@to identifies the endpoint of the lemma in the base text, if necessary.

Status Optional

Datatype `data.pointer`

Values any valid identifier

Note This attribute is only used when the double-end point method of apparatus markup is used, with the encoded apparatus held in a separate file rather than being embedded in-line in the base-text file.

@loc (location) indicates the location of the variation, when the location-referenced method of apparatus markup is used.

Status Mandatory when applicable

Datatype 1–∞ occurrences of `data.word` separated by whitespace

Values Any string containing a canonical reference for the passage to which the variation applies.

Note This attribute is used only when the location-referenced encoding method is used.

Used by model.pPart.transcriptional

May contain

analysis: interp interpGrp span spanGrp

certainty: certainty respons

core: cb gap index lb milestone note pb

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp timeline

spoken: incident kinesic pause shift vocal writing

textcrit: lem rdg rdgGrp wit witDetail

transcr: addSpan damageSpan delSpan fw space

Declaration

```
element app
{
   att.global.attributes,
   attribute type { data.enumerated }?,
   attribute from { data.pointer }?,
   attribute to { data.pointer }?,
   attribute loc { list { data.word+ } }?,
   (
      model.global*,
      ( lem, model.global*, ( wit, model.global* )? )?,
      (
         ( model.rdgLike, model.global*, ( wit, model.global* )? )
       | ( rdgGrp, model.global*, ( wit, model.global* )? )
      )*
   )
}
```

Example

```
<app>
  <lem  wit="#El #Hg">Experience</lem>
  <rdg  wit="#La"  type="substantive">Experiment</rdg>
  <rdg  wit="#Ra2"  type="substantive">Eryment</rdg>
</app>
```

Example

```
<app  type="substantive">
  <rdgGrp  type="subvariants">
    <lem  wit="#El #Hg">Experience</lem>
    <rdg  wit="#Ha4">Experiens</rdg>
  </rdgGrp>
  <rdgGrp  type="subvariants">
    <lem  wit="#Cp #Ld1">Experiment</lem>
    <rdg  wit="#La">Ex<g  ref="#per"/>iment</rdg>
  </rdgGrp>
```

```
  <rdgGrp  type="subvariants">
    <lem>Eriment<wit>[unattested]</wit>
    </lem>
    <rdg  wit="#Ra2">Eryment</rdg>
  </rdgGrp>
</app>
```

Note May contain an optional lemma and one or more readings or reading groups, each associated with witness specifications.

<appInfo> (application information) records information about an application which has edited the TEI file.

Module header — *2. The TEI Header*

Attributes Global attributes only

Used by model.encodingPart

May contain

header: application

Declaration

```
element appInfo { att.global.attributes, model.applicationLike+ }
```

Example

```
<appInfo>
  <application  version="1.24"  ident="Xaira">
    <label>XAIRA Indexer</label>
    <ptr  target="#HD"/>
  </application>
</appInfo>
```

<application> provides information about an application which has acted upon the document.

Module header — *2. The TEI Header*

Attributes att.typed (*@type*, *@subtype*) att.datableatt.datable.w3c (*@period*, *@when*, *@notBefore*, *@notAfter*, *@from*, *@to*) att.datable.iso (*@when-iso*, *@notBefore-iso*, *@notAfter-iso*, *@from-iso*, *@to-iso*)

@ident Supplies an identifier for the application, independent of its version number or display name.

Status **Required**

Datatype `data.name`

@version Supplies a version number for the application, independent of its identifier or display name.

Status **Required**

Datatype

```
token { pattern = "[\d]+[a-z]*[\d]*(\.[\d]+[a-z]*[\d]*){0,3}" }
```

Used by model.applicationLike

May contain

core: desc label p ptr ref

linking: ab

Declaration

```
element application
{
   att.global.attributes,
   att.typed.attributes,
   att.datable.w3c.attributes,
   att.datable.iso.attributes,
   attribute ident { data.name },
   attribute version
   {
      token { pattern = "[\d]+[a-z]*[\d]*(\.[\d]+[a-z]*[\d]*){0,3}" }
   },
   ( model.labelLike+, ( model.ptrLike* | model.pLike* ) )
}
```

Example

```
<appInfo>
  <application version="1.5" ident="ImageMarkupTool1" notAfter="2006-06-01">
    <label>Image Markup Tool</label>
    <ptr target="#P1"/>
    <ptr target="#P2"/>
  </application>
</appInfo>
```

This example shows an appInfo element documenting the fact that version 1.5 of the Image Markup Tool1 application has an interest in two parts of a document which was last saved on June 6 2006. The parts concerned are accessible at the URLs given as target for the two <ptr> elements.

<arc> encodes an arc, the connection from one node to another in a graph.

Module nets — *19. Graphs, Networks, and Trees*

Attributes In addition to global attributes

@from gives the identifier of the node which is adjacent from this arc.

Status **Required**

Datatype `data.pointer`

Values The identifier of a node.

@to gives the identifier of the node which is adjacent to this arc.

Status **Required**

Datatype `data.pointer`

Values The identifier of a node.

Used by graph

May contain

core: label

Declaration

```
element arc
{
   att.global.attributes,
   attribute from { data.pointer },
   attribute to { data.pointer },
   ( label, label? )?
}
```

Example

```
<arc  from="#T3"  to="#T3">
  <label>OLD</label>
  <label>VIEUX</label>
</arc>
```

Note The <arc> element must be used if the arcs are labeled. Otherwise, arcs can be encoded using the **adj**, **adjTo** and **adjFrom** attributes on the <node> tags in the graph. Both <arc> tags and adjacency attributes can be used, but the resulting encoding would be highly redundant.Zero, one, or two children <label> elements may be present. The first occurence of <label> provides a label for the arc; the second provides a second label for the arc, and should be used if a transducer is being encoded.

<argument>

A formal list or prose description of the topics addressed by a subdivision of a text.

Module textstructure — *4. Default Text Structure*

Attributes Global attributes only

Used by opener model.divWrapper model.pLike.front

May contain

analysis: interp interpGrp span spanGrp

certainty: certainty respons

core: bibl biblStruct cb cit desc gap head index l label lb lg list listBibl milestone note p pb q quote said sp stage

dictionaries: entry entryFree superEntry

drama: camera caption castList move sound tech view

figures: figure table

header: biblFull

iso-fs: fLib fs fvLib

linking: ab alt altGrp anchor join joinGrp link linkGrp timeline

msdescription: msDesc

namesdates: listNym listOrg listPerson listPlace

nets: eTree forest forestGrp graph tree

spoken: incident kinesic pause shift u vocal writing

tagdocs: classSpec eg egXML elementSpec listRef macroSpec moduleRef moduleSpec schemaSpec specGrp specGrpRef

textcrit: listWit witDetail

textstructure: floatingText

transcr: addSpan damageSpan delSpan fw space

Declaration

```
element argument
{
   att.global.attributes,
   ( ( model.global | model.headLike )*, ( ( model.common ), model.global* )+ )
}
```

Example

```
<argument>
  <p>Monte Video — Maldonado — Excursion
     to R Polanco — Lazo and Bolas — Partridges —
     Absence of Trees — Deer — Capybara, or River Hog —
     Tucutuco — Molothrus, cuckoo-like habits — Tyrant
     Flycatcher — Mocking-bird — Carrion Hawks —
     Tubes formed by Lightning — House struck</p>
</argument>
```

Note Often contains either a list or a paragraph

<att> (attribute) contains the name of an attribute appearing within running text.

Module tagdocs — *22. Documentation Elements*

Attributes In addition to global attributes

@scheme supplies an identifier for the scheme in which this name is defined.

Status Optional

Datatype `data.enumerated`

Sample values include: **TEI** (text encoding initiative) this attribute is part of the TEI scheme. **[Default]**

DBK (docbook) this attribute is part of the Docbook scheme.

XX (unknown) this attribute is part of an unknown scheme.

Used by model.phrase.xml

May contain Character data only

Declaration

```
element att
{
   att.global.attributes,
   attribute scheme { data.enumerated }?,
   text
}
```

Example

```
<p>The TEI defines six <soCalled>global</soCalled> attributes;
their names are <att>xml:id</att>, <att>rend</att>,
<att>xml:lang</att>, <att>n</att>, <att>xml:space</att>, and
<att>xml:base</att>; <att  scheme="XX">style</att> is not among
them.</p>
```

Note A namespace prefix may be used in order to specify the scheme as an alternative to specifying it via the scheme attribute: it takes precedence

---

<attDef> (attribute definition) contains the definition of a single attribute.

Module tagdocs — *22. Documentation Elements*

Attributes att.identified (*@ident*, *@predeclare*, *@module*, *@mode*)

@usage specifies the optionality of an attribute or element.

Status Optional

Legal values are: **req** (required)

mwa (mandatory when applicable)

rec (recommended)

rwa (recommended when applicable)

opt (optional) **[Default]**

@ns (namespace) specifies the namespace to which this attribute belongs

Status Optional

Datatype `data.namespace`

Used by attList

May contain

core: desc gloss

tagdocs: altIdent datatype defaultVal equiv exemplum remarks valDesc valList

Declaration

```
element attDef
{
   att.global.attributes,
   att.identified.attributes,
   attribute usage { "req" | "mwa" | "rec" | "rwa" | "opt" }?,
   attribute ns { data.namespace }?,
   (
      model.glossLike*,
      datatype?,
      defaultVal?,
      ( valList | valDesc )?,
      exemplum*,
      remarks*
   )
}
```

Example

```
<attDef  usage="rec"  ident="type">
  <desc>specifies a name conventionally used for this
     level of subdivision, e.g. <val>act</val>, <val>volume</val>,
  <val>book</val>, <val>section</val>, <val>canto</val>, etc.</desc>
</attDef>
```

<attList> contains documentation for all the attributes associated with this element, as a series of <attDef> elements.

Module tagdocs — *22. Documentation Elements*

Attributes In addition to global attributes

@org (organization) specifies whether all the attributes in the list are available (org="group") or only one of them (org="choice")

Status Optional

Legal values are: **group** grouped **[Default]**

choice alternated

Used by attList classSpec elementSpec

May contain

tagdocs: attDef attList attRef

Declaration

```
element attList
{
   att.global.attributes,
   attribute org { "group" | "choice" }?,
   ( attRef | attDef | attList )+
}
```

Example

```
<attList>
  <attDef  ident="type"  usage="opt">
    <equiv/>
    <desc>type of schema</desc>
    <datatype>
      <rng:ref  name="data.enumerated"/>
    </datatype>
  </attDef>
</attList>
```

<attRef/> (attribute pointer) points to the definition of an attribute or group of attributes.

Module tagdocs — *22. Documentation Elements*

Attributes In addition to global attributes

@name the name of the pattern defining the attribute(s)

Status **Required**

Datatype `data.word`

Used by attList

May contain Empty element

Declaration

```
element attRef { att.global.attributes, attribute name { data.word }, empty }
```

Example

```
<attRef name="att.global.attribute.xml:id"/>
```

<author> in a bibliographic reference, contains the name of the author(s), personal or corporate, of a work; the primary statement of responsibility for any bibliographic item.

Module core — *3. Elements Available in All TEI Documents*

Attributes Global attributes only

Used by analytic monogr msItemStruct titleStmt model.respLike

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address binaryObject cb choice corr date del distinct email emph expan foreign gap gloss graphic hi index lb measure measureGrp mentioned milestone name note num orig pb ptr ref reg rs sic soCalled term time title unclear

dictionaries: lang oRef oVar pRef pVar

figures: formula

gaiji: g

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident specDesc specList tag val

textcrit: app witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element author { att.global.attributes, macro.phraseSeq }
```

Example

```
<author>British Broadcasting Corporation</author>
<author>La Fayette, Marie Madeleine Pioche de la Vergne,
comtesse de (1634–1693)</author>
```

Source: [57]

Note Particularly where cataloguing is likely to be based on the content of the header, it is advisable to use generally recognized authority lists for the exact form of personal names.In the case of a broadcast, use this element for the name of the company or network which broadcasts the program.

<authority> (release authority) supplies the name of a person or other agency responsible for making an electronic file available, other than a publisher or distributor.

Module header — *2. The TEI Header*

Attributes Global attributes only

Used by model.publicationStmtPart

May contain

analysis: interp interpGrp span spanGrp

certainty: certainty respons

core: abbr address cb choice date distinct email emph expan foreign gap gloss index lb measure measureGrp mentioned milestone name note num pb ptr ref rs soCalled term time title

dictionaries: lang

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident tag val

textcrit: witDetail

transcr: addSpan am damageSpan delSpan ex fw handShift space subst

Declaration

```
element authority { att.global.attributes, macro.phraseSeq.limited }
```

Example

```
<authority>John Smith</authority>
```

<availability> supplies information about the availability of a text, for example any restrictions on its use or distribution, its copyright status, etc.

Module header — *2. The TEI Header*

Attributes att.declarable (*@default*)

@status supplies a code identifying the current availability of the text.

Status Optional

Legal values are: **free** the text is freely available.

unknown the status of the text is unknown. **[Default]**

restricted the text is not freely available.

Used by adminInfo model.publicationStmtPart

May contain

core: p

linking: ab

Declaration

```
element availability
{
   att.global.attributes,
   att.declarable.attributes,
   attribute status { "free" | "unknown" | "restricted" }?,
   model.pLike+
}
```

Example

```
<availability status="restricted">
  <p>Available for academic research purposes only.</p>
</availability>
<availability status="free">
  <p>In the public domain</p>
</availability>
<availability status="restricted">
  <p>Available under licence from the publishers.</p>
</availability>
```

Note A consistent format should be adopted

<back>

(back matter) contains any appendixes, etc. following the main part of a text.

Module textstructure — *4. Default Text Structure*

Attributes att.declaring (*@decls*)

Used by facsimile floatingText text

May contain

analysis: interp interpGrp span spanGrp

certainty: certainty respons

core: cb divGen gap head index lb milestone note pb

drama: castList epilogue performance prologue set

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp timeline

spoken: incident kinesic pause shift vocal writing

textcrit: witDetail

textstructure: argument byline closer div div1 docAuthor docDate docEdition docImprint docTitle epigraph postscript signed titlePage titlePart trailer

transcr: addSpan damageSpan delSpan fw space

Declaration

```
element back
{
   att.global.attributes,
   att.declaring.attributes,
```

```
   (
      ( model.frontPart | model.pLike.front | model.global )*,
      (
         (
            (
               ( model.div1Like ),
               ( model.frontPart | model.div1Like | model.global )*
            )
          | (
               ( model.divLike ),
               ( model.frontPart | model.divLike | model.global )*
            )
         )?
      ),
      ( ( ( model.divBottomPart ), ( model.divBottomPart | model.global )* )? )
   )
}
```

Example

```
<back>
  <div1  type="appendix">
    <head>The Golden Dream or, the Ingenuous Confession</head>
    <p>To shew the Depravity of human Nature
    </p>
  </div1>
  <div1  type="epistle">
    <head>A letter from the Printer, which he desires
        may be inserted</head>
    <salute>Sir.</salute>
    <p>I have done with your Copy, so you may return
        it to the Vatican, if you please
    </p>
  </div1>
  <div1  type="advert">
    <head>The Books usually read by the Scholars
        of Mrs Two-Shoes are these and are sold at Mr
        Newbery's at the Bible and Sun in St Paul's
        Church-yard.</head>
    <list>
      <item  n="1">The Christmas Box, Price 1d.</item>
      <item  n="2">The History of Giles Gingerbread, 1d.</item>
      <item  n="42">A Curious Collection of Travels, selected
            from the Writers of all Nations, 10 Vol, Pr.  bound 1l.</item>
    </list>
  </div1>
  <div1  type="advert">
    <head>
      <hi  rend="center">By the KING's Royal Patent,</hi>
        Are sold by J. NEWBERY, at the Bible and Sun in
        St.  Paul's Church-Yard.</head>
    <list>
      <item  n="1">Dr.  James's Powders for Fevers, the
            Small-Pox, Measles, Colds, &c.  2s.  6d</item>
      <item  n="2">Dr.  Hooper's Female Pills, 1s.</item>
    </list>
  </div1>
</back>
```

Note The content model of back matter is identical to that of front matter, reflecting the facts of cultural history.

<bibl> (bibliographic citation) contains a loosely-structured bibliographic citation of which the sub-components may or may not be explicitly tagged.

Module core — *3. Elements Available in All TEI Documents*

Attributes att.declarable (*@default*) att.typed (*@type*, *@subtype*)

Used by msItemStruct model.biblLike model.msItemPart model.personPart

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address author biblScope cb choice corr date del distinct editor email emph expan foreign gap gloss hi index lb measure measureGrp meeting mentioned milestone name note num orig pb ptr pubPlace publisher ref reg relatedItem respStmt rs series sic soCalled term time title unclear

dictionaries: lang

gaiji: g

header: distributor edition extent idno

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: depth height msIdentifier width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: code ident

textcrit: app witDetail

transcr: addSpan am damage damageSpan delSpan ex fw restore space subst supplied

Declaration

```
element bibl
{
   att.global.attributes,
   att.declarable.attributes,
   att.typed.attributes,
   (
      text
    | model.gLike    | model.highlighted    | model.pPart.data    | model.pPart.edit    | model.segLike    | mod
}
```

Example

```
<bibl>Blain, Clements and Grundy: Feminist Companion to Literature in
English (Yale, 1990)</bibl>
```

Example

```
<bibl>
  <title  level="a">The Interesting story of the Children in the Wood</title>.
In <author>Victor E Neuberg</author>, <title>The Penny Histories</title>.
<publisher>OUP</publisher>
  <date>1968</date>.

</bibl>
```

Note Contains phrase-level elements, together with any combination of elements from the biblPart class

---

<biblFull> (fully-structured bibliographic citation) contains a fully-structured bibliographic citation, in which all components of the TEI file description are present.

Module header — *2. The TEI Header*

Attributes att.declarable (*@default*)

Used by model.biblLike

May contain

header: editionStmt extent notesStmt publicationStmt seriesStmt sourceDesc titleStmt

Declaration

```
element biblFull
{
   att.global.attributes,
   att.declarable.attributes,
   (
      (
         titleStmt,
         editionStmt?,
         extent?,
         publicationStmt,
         seriesStmt?,
         notesStmt?
      ),
      sourceDesc*
   )
}
```

Example

```
<biblFull>
  <titleStmt>
    <title>The Feminist Companion to Literature in English:
        women writers from the middle ages to the present</title>
    <author>Blain, Virginia</author>
    <author>Clements, Patricia</author>
    <author>Grundy, Isobel</author>
  </titleStmt>
  <editionStmt>
    <edition>UK edition</edition>
  </editionStmt>
  <extent>1231 pp</extent>
  <publicationStmt>
```

```
    <publisher>Yale University Press</publisher>
    <pubPlace>New Haven and London</pubPlace>
    <date>1990</date>
  </publicationStmt>
  <sourceDesc>
    <p>No source: this is an original work</p>
  </sourceDesc>
</biblFull>
```

<biblScope> (scope of citation) defines the scope of a bibliographic reference, for example as a list of page numbers, or a named subdivision of a larger work.

Module core — *3. Elements Available in All TEI Documents*

Attributes In addition to global attributes

@type identifies the type of information conveyed by the element, e.g. **pages**, **volume**.

Status Optional

Datatype `data.enumerated`

Suggested values include: **vol** (volume) the element contains a volume number.

issue the element contains an issue number, or volume and issue numbers.

pp (pages) the element contains a page number or page range.

chap (chapter) the element contains a chapter indication (number and/or title)

part the element identifies a part of a book or collection.

Used by monogr series model.imprintPart

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address binaryObject cb choice corr date del distinct email emph expan foreign gap gloss graphic hi index lb measure measureGrp mentioned milestone name note num orig pb ptr ref reg rs sic soCalled term time title unclear

dictionaries: lang oRef oVar pRef pVar

figures: formula

gaiji: g

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident specDesc specList tag val

textcrit: app witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element biblScope
{
   att.global.attributes,
   attribute type { "vol" | "issue" | "pp" | "chap" | "part" | xsd:Name }?,
   macro.phraseSeq}
```

Example

```
<biblScope>pp 12–34</biblScope>
<biblScope  type="vol">II</biblScope>
<biblScope  type="pp">12</biblScope>
```

<biblStruct> (structured bibliographic citation) contains a structured bibliographic citation, in which only bibliographic sub-elements appear and in a specified order.

Module core — *3. Elements Available in All TEI Documents*

Attributes att.declarable (*@default*) att.typed (*@type*, *@subtype*)

Used by model.biblLike

May contain

core: analytic monogr note relatedItem series

header: idno

textcrit: witDetail

Declaration

```
element biblStruct
{
   att.global.attributes,
   att.declarable.attributes,
   att.typed.attributes,
   (
      analytic*,
      ( monogr, series* )+,
      ( model.noteLike | idno | relatedItem )*
   )
}
```

Example

```
<biblStruct>
  <monogr>
    <author>Blain, Virginia</author>
    <author>Clements, Patricia</author>
    <author>Grundy, Isobel</author>
    <title>The Feminist Companion to Literature in
        English: women writers from the middle ages
        to the present</title>
```

```
    <edition>first edition</edition>
    <imprint>
      <publisher>Yale University Press</publisher>
      <pubPlace>New Haven and London</pubPlace>
      <date>1990</date>
    </imprint>
  </monogr>
</biblStruct>
```

<bicond> (bi-conditional feature-structure constraint) defines a biconditional feature-structure constraint; both consequent and antecedent are specified as feature structures or groups of feature structures; the constraint is satisfied if both subsume a given feature structure, or if both do not.

Module iso-fs — *18. Feature Structures*

Attributes Global attributes only

Used by fsConstraints

May contain

iso-fs: f fs iff

Declaration

```
element bicond { att.global.attributes, ( ( fs | f ), iff, ( fs | f ) ) }
```

Example

```
<bicond>
  <fs>
    <f name="FOO">
      <symbol value="42"/>
    </f>
  </fs>
  <iff/>
  <fs>
    <f name="BAR">
      <binary value="true"/>
    </f>
  </fs>
</bicond>
```

<binary/> (binary value) represents the value part of a feature-value specification which can contain either of exactly two possible values.

Module iso-fs — *18. Feature Structures*

Attributes In addition to global attributes

@value supplies a binary value.

Status **Required**

Datatype `data.truthValue`

Values a string representing a binary value (true or false, 0 or 1) .

Used by model.featureVal.single

May contain Empty element

Declaration

```
element binary
{
   att.global.attributes,
   attribute value { data.truthValue },
   empty
}
```

Example

```
<f name="strident">
  <binary value="true"/>
</f>
<f name="exclusive">
  <binary value="false"/>
</f>
```

Note The value attribute may take any value permitted for attributes of the W3C datatype Boolean: this includes for example the strings `true` or `1` which are equivalent.

<binaryObject> provides encoded binary data representing an inline graphic or other object.

Module core — *3. Elements Available in All TEI Documents*

Attributes att.internetMedia (*@mimeType*)

@width The display width of the object

Status Mandatory when applicable

Datatype `data.outputMeasurement`

@height The display height of the object

Status Mandatory when applicable

Datatype `data.outputMeasurement`

@scale A scale factor to be applied to the object to make it the desired display size

Status Mandatory when applicable

Datatype `data.numeric`

@encoding The encoding used to encode the binary data. If not specified, this is assumed to be Base64.

Status Optional

Datatype 1–∞ occurrences of `data.word` separated by whitespace

Used by model.graphicLike model.titlepagePart

May contain Character data only

Declaration

```
element binaryObject
{
   att.global.attributes,
   att.internetMedia.attributes,
   attribute width { data.outputMeasurement }?,
   attribute height { data.outputMeasurement }?,
   attribute scale { data.numeric }?,
```

```
   attribute encoding { list { data.word+ } }?,
   text
}
```

Example

```
<binaryObject mimeType="image/gif">R0lGODdhMAAwAPAAAAAAAP///ywAAAAAMAAwAAAC8IyPqcvt3wCcDkiLc7C0qwy
GHhSWpjQu5yqmCYsapyuvUUlvONmOZtfzgFzByTB10QgxOR0TqBQejhRNzOfkVJ
+5YiUqrXF5Y5lKh/DeuNcP5yLWGsEbtLiOSpa/TPg7JpJHxyendzWTBfX0cxOnK
PjgBzi4diinWGdkF8kjdfnycQZXZeYGejmJlZeGl9i2icVqaNVailT6F5iJ90m6
mvuTS4OK05M0vDk0Q4XUtwvKOzrcd3iq9uisF81M1OIcR7lEewwcLp7tuNNkM3u
Nna3F2JQFo97Vriy/Xl4/f1cf5VWzXyym7PH hhx4dbgYKAAA7</binaryObject>
```

Note The MIME media type specified on the **mimeType** attribute should describe the object after it has been decoded.

<binding> contains a description of one binding, i.e. type of covering, boards, etc. applied to a manuscript.

Module msdescription — *10. Manuscript Description*

Attributes att.datableatt.datable.w3c (*@period*, *@when*, *@notBefore*, *@notAfter*, *@from*, *@to*)
att.datable.iso (*@when-iso*, *@notBefore-iso*, *@notAfter-iso*, *@from-iso*, *@to-iso*)

@contemporary specifies whether or not the binding is contemporary with the majority of its contents

Status Optional

Datatype `data.xTruthValue`

Note The value **true** indicates that the binding is contemporaneous with its contents; the value **false** that it is not. The value **unknown** should be used when the date of either binding or manuscript is unknown

Used by bindingDesc

May contain

core: p

linking: ab

msdescription: decoNote

Declaration

```
element binding
{
   att.global.attributes,
   att.datable.w3c.attributes,
   att.datable.iso.attributes,
   attribute contemporary { data.xTruthValue }?,
   ( model.pLike | decoNote )+
}
```

Example

```
<binding contemporary="true">
  <p>Contemporary blind stamped leather over wooden
     boards with evidence of a fore edge clasp closing
```

```
        to the back cover.</p>
    </binding>
```

Example

```
    <bindingDesc>
      <binding  contemporary="false">
        <p>Quarter bound by the Phillipps' binder, Bretherton,
            with his sticker on the front pastedown.</p>
      </binding>
      <binding  contemporary="false">
        <p>Rebound by an unknown 19th c. company; edges cropped and
            gilt.</p>
      </binding>
    </bindingDesc>
```

<bindingDesc> (binding description) describes the present and former bindings of a manuscript, either as a series of paragraphs or as a series of distinct <binding> elements, one for each binding of the manuscript.

Module msdescription — *10. Manuscript Description*

Attributes Global attributes only

Used by model.physDescPart

May contain

core: p

linking: ab

msdescription: binding decoNote

Declaration

```
element bindingDesc
{
   att.global.attributes,
   ( ( model.pLike | decoNote )+ | binding+ )
}
```

Example

```
    <bindingDesc>
      <p>Sewing not visible; tightly rebound over
         19th-cent. pasteboards, reusing panels of 16th-cent. brown leather with
         gilt tooling à la fanfare, Paris c. 1580-90, the centre of each
         cover inlaid with a 17th-cent. oval medallion of red morocco tooled in
         gilt (perhaps replacing the identifying mark of a previous owner); the
         spine similarly tooled, without raised bands or title-piece; coloured
         endbands; the edges of the leaves and boards gilt.Boxed.</p>
    </bindingDesc>
```

<birth> contains information about a person's birth, such as its date and place.

Module namesdates — *13. Names, Dates, People, and Places*

Attributes att.editLike (*@cert*, *@resp*, *@evidence*, *@source*, *@precision*) att.datableatt.datable.w3c (*@period*, *@when*, *@notBefore*, *@notAfter*, *@from*, *@to*) att.datable.iso (*@when-iso*, *@notBefore-iso*, *@notAfter-iso*, *@from-iso*, *@to-iso*) att.naming (*@key*, *@ref*, *@nymRef*)

Used by model.persEventLike

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address binaryObject cb choice corr date del distinct email emph expan foreign gap gloss graphic hi index lb measure measureGrp mentioned milestone name note num orig pb ptr ref reg rs sic soCalled term time title unclear

dictionaries: lang oRef oVar pRef pVar

figures: formula

gaiji: g

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident specDesc specList tag val

textcrit: app witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element birth
{
   att.global.attributes,
   att.editLike.attributes,
   att.datable.w3c.attributes,
   att.datable.iso.attributes,
   att.naming.attributes,
   macro.phraseSeq}
```

Example

```
<birth>Before 1920, Midlands region.</birth>
```

Example

```
<birth  when="1960-12-10">In a small cottage near
<name  type="place">Aix-la-Chapelle</name>,
early in the morning of <date>10 Dec 1960</date>
</birth>
```

<bloc> contains the name of a geo-political unit consisting of two or more nation states or countries.

Module namesdates — *13. Names, Dates, People, and Places*

Attributes att.naming (*@key*, *@ref*, *@nymRef*) att.typed (*@type*, *@subtype*) att.datableatt.datable.w3c (*@period*, *@when*, *@notBefore*, *@notAfter*, *@from*, *@to*) att.datable.iso (*@when-iso*, *@notBefore-iso*, *@notAfter-iso*, *@from-iso*, *@to-iso*)

Used by model.placeNamePart

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address binaryObject cb choice corr date del distinct email emph expan foreign gap gloss graphic hi index lb measure measureGrp mentioned milestone name note num orig pb ptr ref reg rs sic soCalled term time title unclear

dictionaries: lang oRef oVar pRef pVar

figures: formula

gaiji: g

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident specDesc specList tag val

textcrit: app witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element bloc
{
   att.global.attributes,
   att.naming.attributes,
   att.typed.attributes,
   att.datable.w3c.attributes,
   att.datable.iso.attributes,
   macro.phraseSeq}
```

Example

```
<bloc type="union">the European Union</bloc>
<bloc type="continent">Africa</bloc>
```

<body> (text body) contains the whole body of a single unitary text, excluding any front or back matter.

Module textstructure — *4. Default Text Structure*

Attributes att.declaring (*@decls*)

Used by floatingText text

May contain

analysis: interp interpGrp span spanGrp

certainty: certainty respons

core: bibl biblStruct cb cit desc divGen gap head index l label lb lg list listBibl meeting milestone note p pb q quote said sp stage

dictionaries: entry entryFree superEntry

drama: camera caption castList move sound tech view

figures: figure table

header: biblFull

iso-fs: fLib fs fvLib

linking: ab alt altGrp anchor join joinGrp link linkGrp timeline

msdescription: msDesc

namesdates: listNym listOrg listPerson listPlace

nets: eTree forest forestGrp graph tree

spoken: incident kinesic pause shift u vocal writing

tagdocs: classSpec eg egXML elementSpec listRef macroSpec moduleRef moduleSpec schemaSpec specGrp specGrpRef

textcrit: listWit witDetail

textstructure: argument byline closer dateline div div1 docAuthor docDate epigraph floatingText opener postscript salute signed trailer

transcr: addSpan damageSpan delSpan fw space

Declaration

```
element body
{
   att.global.attributes,
   att.declaring.attributes,
   (
      model.global*,
      ( ( model.divTop ), ( model.global | model.divTop )* )?,
      ( ( model.divGenLike ), ( model.global | model.divGenLike )* )?,
      (
         ( ( model.divLike ), ( model.global | model.divGenLike )* )+
       | ( ( model.div1Like ), ( model.global | model.divGenLike )* )+
       | (
            ( ( model.common ), model.global* )+,
            (
               ( ( model.divLike ), ( model.global | model.divGenLike )* )+
             | ( ( model.div1Like ), ( model.global | model.divGenLike )* )+
            )?
         )
      ),
      ( ( model.divBottom ), model.global* )*
   )
}
```

Example

```
<body>
  <l>Nu scylun hergan hefaenricaes uard</l>
  <l>metudæs maecti end his modgidanc</l>
  <l>uerc uuldurfadur sue he uundra gihuaes</l>
  <l>eci dryctin or astelidæ</l>
  <l>he aerist scop aelda barnum</l>
  <l>heben til hrofe haleg scepen.</l>
  <l>tha middungeard moncynnæs uard</l>
  <l>eci dryctin æfter tiadæ</l>
  <l>firum foldu frea allmectig</l>
  <trailer>primo cantauit Cædmon istud carmen.
  </trailer>
</body>
```

<broadcast> describes a broadcast used as the source of a spoken text.

Module spoken — *8. Transcriptions of Speech*

Attributes att.declarable (*@default*)

Used by model.recordingPart

May contain

core: bibl biblStruct p

header: biblFull

linking: ab

msdescription: msDesc

spoken: recording

Declaration

```
element broadcast
{
   att.global.attributes,
   att.declarable.attributes,
   ( model.pLike+ | model.biblLike | recording )
}
```

Example

```
<broadcast>
  <bibl>
    <author>Radio Trent</author>
    <title>Gone Tomorrow</title>
    <respStmt>
      <resp>Presenter</resp>
      <name>Tim Maby</name>
    </respStmt>
    <respStmt>
      <resp>Producer</resp>
      <name>Mary Kerr</name>
    </respStmt>
    <date  when="1989-06-12T12:30:00">12 June 89, 1230 pm</date>
```

```
  </bibl>
</broadcast>
```

<byline> contains the primary statement of responsibility given for a work on its title page or at the head or end of the work.

Module textstructure — *4. Default Text Structure*

Attributes Global attributes only

Used by opener model.divWrapper model.titlepagePart model.pLike.front

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address binaryObject cb choice corr date del distinct email emph expan foreign gap gloss graphic hi index lb measure measureGrp mentioned milestone name note num orig pb ptr ref reg rs sic soCalled term time title unclear

dictionaries: lang oRef oVar pRef pVar

figures: formula

gaiji: g

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident specDesc specList tag val

textcrit: app witDetail

textstructure: docAuthor

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element byline
{
   att.global.attributes,
   ( text | model.gLike | model.phrase | docAuthor | model.global )*
}
```

Example

```
<byline>Written by a CITIZEN who continued all the
while in London. Never made publick before.</byline>
```

Example

```
<byline>Written from her own MEMORANDUMS</byline>
```

Example

```
<byline>By George Jones, Political Editor, in Washington</byline>
```

Example

```
<dateline>Zagreb:</dateline>
<byline>de notre envoyé spécial.</byline>
```

Example

```
<byline>BY
<docAuthor>THOMAS PHILIPOTT,</docAuthor>
Master of Arts,
(Somtimes)
Of Clare-Hall in Cambridge.</byline>
```

Note The byline on a title page may include either the name or a description for the document's author. Where the name is included, it may optionally be tagged using the <docAuthor> element.

<c> (character) represents a character.

Module analysis — *17. Simple Analytic Mechanisms*

Attributes att.segLike (*@type*, *@function*, *@part*) att.metrical (*@met*, *@real*, *@rhyme*)

Used by model.segLike

May contain

gaiji: g

Declaration

```
element c
{
   att.global.attributes,
   att.segLike.attributes,
   att.metrical.attributes,
   macro.xtext}
```

Example

```
<c  type="punctuation">?</c>
```

Note Contains a single character or a sequence of graphemes to be treated as a single character.

<cRefPattern> (canonical reference pattern) specifies an expression and replacement pattern for transforming a canonical reference into a URI.

Module header — *2. The TEI Header*

Attributes In addition to global attributes

@matchPattern specifies a regular expression against which the values of **cRef** attributes can be matched.

Status **Required**

Datatype `data.pattern`

Values must be a regular expression according to the W3C XML Schema Language

Note Parenthesised groups are used not only for establishing order of precedence and atoms for quantification, but also for creating subpatterns to be referenced by the **replacementPattern** attribute.

@replacementPattern specifies a 'replacement pattern' which, once subpattern substitution has been performed, provides a URI.

Status **Required**

Datatype `text`

Values Should be the skeleton of a relative or absolute URI, with references to groups in the **matchPattern**.

Note The strings '$1' through '$9' are references to the corresponding group in the regular expression specified by **matchPattern** (counting open parenthesis, left to right). Processors are expected to replace them with whatever matched the corresponding group in the regular expression.If a digit preceded by a dollar sign is needed in the actual replacement pattern (as opposed to being used as a back reference), the dollar sign must be written as `%24`.

Used by refsDecl

May contain

core: p

linking: ab

Declaration

```
element cRefPattern
{
   att.global.attributes,
   attribute matchPattern { data.pattern },
   attribute replacementPattern { text },
   model.pLike*
}
```

Example

```
<cRefPattern
   matchPattern="([1-9A-Za-z]+)\s+([0-9]+):([0-9]+)"
   replacementPat-
tern="#xpath(//div[@type='book'][@n='$1']/div[@type='chap'][@n='$2']/div[@type='verse'][@n='$3'])"/>
```

Note The result of the substitution may be either an absolute or a relative URI reference. In the latter case it is combined with the value of **xml:base** in force at the place where the **cRef** attribute occurs to form an absolute URI in the usual manner as prescribed by XML Base.

<caesura/> marks the point at which a metrical line may be divided.

Module verse — *6. Verse*

Attributes Global attributes only

Used by model.lPart

May contain Empty element

Declaration

```
element caesura { att.global.attributes, empty }
```

Example

```
<l>Hwæt we Gar-Dena <caesura/> in gear-dagum</l>
<l>þeod-cyninga <caesura/> þrym gefrunon,</l>
<l>hy ða æþelingas <caesura/> ellen fremedon.</l>
```

Source: [15]

<camera> describes a particular camera angle or viewpoint in a screen play.

Module drama — *7. Performance Texts*

Attributes att.typed (*@type*, *@subtype*)

Used by model.stageLike

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address bibl biblStruct binaryObject cb choice cit corr date del desc distinct email emph expan foreign gap gloss graphic hi index label lb list listBibl measure measureGrp mentioned milestone name note num orig pb ptr q quote ref reg rs said sic soCalled stage term time title unclear

dictionaries: lang oRef oVar pRef pVar

drama: camera caption castList move sound tech view

figures: figure formula table

gaiji: g

header: biblFull

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material msDesc origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName listNym listOrg listPerson listPlace nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att classSpec code eg egXML elementSpec gi ident listRef macroSpec moduleRef moduleSpec specDesc specGrp specGrpRef specList tag val

textcrit: app listWit witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element camera
```

```
{
   att.global.attributes,
   att.typed.attributes,
   macro.paraContent}
```

Example

```
<view>George glances at the window--and freezes.
<camera  type="cut">New angle--shock cut</camera>
Out the window the body of a dead man suddenly slams into frame
</view>
```

<caption> contains the text of a caption or other text displayed as part of a film script or screenplay.

Module drama — *7. Performance Texts*

Attributes Global attributes only

Used by model.stageLike

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address bibl biblStruct binaryObject cb choice cit corr date del desc distinct email emph expan foreign gap gloss graphic hi index label lb list listBibl measure measureGrp mentioned milestone name note num orig pb ptr q quote ref reg rs said sic soCalled stage term time title unclear

dictionaries: lang oRef oVar pRef pVar

drama: camera caption castList move sound tech view

figures: figure formula table

gaiji: g

header: biblFull

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material msDesc origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName listNym listOrg listPerson listPlace nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att classSpec code eg egXML elementSpec gi ident listRef macroSpec moduleRef moduleSpec specDesc specGrp specGrpRef specList tag val

textcrit: app listWit witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element caption { att.global.attributes, macro.paraContent }
```

Example

```
<camera>Zoom in to overlay showing some stock film of hansom cabs
galloping past</camera>
<caption>London, 1895.</caption>
<caption>The residence of Mr Oscar Wilde.</caption>
<sound>Suitably classy music starts.</sound>
<view>Mix through to Wilde's drawing room. A crowd of suitably
dressed folk are engaged in typically brilliant conversation,
laughing affectedly and drinking champagne.</view>
<sp>
  <speaker>Prince of Wales</speaker>
  <p>My congratulations, Wilde. Your latest play is a great success.
  </p>
</sp>
```

Note A specialized form of stage direction.

<case> contains grammatical case information given by a dictionary for a given form.

Module dictionaries — *9. Dictionaries*

Attributes att.lexicographic (*@expand*, *@norm*, *@split*, *@value*, *@orig*, *@location*, *@mergedIn*, *@opt*)

Used by model.entryPart model.morphLike

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address bibl biblStruct binaryObject cb choice cit corr date del desc distinct email emph expan foreign gap gloss graphic hi index label lb list listBibl measure measureGrp mentioned milestone name note num orig pb ptr q quote ref reg rs said sic soCalled stage term time title unclear

dictionaries: lang oRef oVar pRef pVar

drama: camera caption castList move sound tech view

figures: figure formula table

gaiji: g

header: biblFull

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material msDesc origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName listNym listOrg listPerson listPlace nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att classSpec code eg egXML elementSpec gi ident listRef macroSpec moduleRef moduleSpec specDesc specGrp specGrpRef specList tag val

textcrit: app listWit witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element case
{
   att.global.attributes,
   att.lexicographic.attributes,
   macro.paraContent}
```

Example Taken from *Wörterbuch der Deutschen Sprache. Veranstaltet und herausgegeben von Joachim Heinrich Campe. Erster Theil. A - bis - E.* (Braunschweig 1807. In der Schulbuchhandlung):

Das Evangelium, des Evangelii, ...

```
<entry>
  <form type="lemma">
    <gramGrp>
      <pos value="noun"/>
      <gen value="n"/>
    </gramGrp>
    <form type="determiner">
      <orth>Das</orth>
    </form>
    <form type="headword">
      <orth>Evangelium</orth>,</form>
  </form>
  <form type="inflected">
    <gramGrp>
      <case value="genitiv"/>
      <number value="singular"/>
    </gramGrp>
    <form type="determiner">
      <orth>des</orth>
    </form>
    <form type="headword">
      <orth>
        <oVar>Evangelii</oVar>,</orth>
    </form>
  </form>
</entry>
```

Note May contain character data and phrase-level elements. Typical values will be of the form *nominative*, *accusative*, *dative*, *genitive*, etc.This element is synonymous with **<gram type="case">**.

<castGroup> (cast list grouping) groups one or more individual castItem elements within a cast list.

Module drama — *7. Performance Texts*

Attributes Global attributes only

Used by castGroup castList

May contain

analysis: interp interpGrp span spanGrp

certainty: certainty respons

core: cb gap head index lb milestone note pb

drama: castGroup castItem roleDesc

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp timeline

spoken: incident kinesic pause shift vocal writing

textcrit: witDetail

textstructure: trailer

transcr: addSpan damageSpan delSpan fw space

Declaration

```
element castGroup
{
   att.global.attributes,
   (
      ( model.global | model.headLike )*,
      ( ( castItem | castGroup | roleDesc ), model.global* )+,
      ( trailer, model.global* )?
   )
}
```

Example

```
<castGroup  rend="braced">
  <castItem>
    <role>Walter</role>
    <actor>Mr Frank Hall</actor>
  </castItem>
  <castItem>
    <role>Hans</role>
    <actor>Mr F.W. Irish</actor>
  </castItem>
  <roleDesc>friends of Mathias</roleDesc>
</castGroup>
```

Note The **rend** attribute may be used, as here, to indicate whether the grouping is indicated by a brace, whitespace, font change, etc. Note that in this example the role description 'friends of Mathias' is understood to apply to both roles equally.

<castItem> (cast list item) contains a single entry within a cast list, describing either a single role or a list of non-speaking roles.

Module drama — *7. Performance Texts*

Attributes In addition to global attributes

@type characterizes the cast item.

Status Optional

Legal values are: **role** the item describes a single role. **[Default]**

list the item describes a list of non-speaking roles.

Used by castGroup castList

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address binaryObject cb choice corr date del distinct email emph expan foreign gap gloss graphic hi index lb measure measureGrp mentioned milestone name note num orig pb ptr ref reg rs sic soCalled term time title unclear

dictionaries: lang oRef oVar pRef pVar

drama: actor role roleDesc

figures: formula

gaiji: g

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident specDesc specList tag val

textcrit: app witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element castItem
{
   att.global.attributes,
   attribute type { "role" | "list" }?,
   ( text | model.gLike | model.castItemPart | model.phrase | model.global )*
}
```

Example

```
<castItem>
  <role>Player</role>
  <actor>Mr Milward</actor>
</castItem>
```

Example

```
<castItem  type="list">Constables, Drawer, Turnkey, etc.</castItem>
```

<castList> (cast list) contains a single cast list or dramatis personae.

Module drama — *7. Performance Texts*

Attributes Global attributes only

Used by model.inter model.frontPart.drama

May contain

analysis: interp interpGrp span spanGrp

certainty: certainty respons

core: bibl biblStruct cb cit desc gap head index l label lb lg list listBibl meeting milestone note p pb q quote said sp stage

dictionaries: entry entryFree superEntry

drama: camera caption castGroup castItem castList move sound tech view

figures: figure table

header: biblFull

iso-fs: fLib fs fvLib

linking: ab alt altGrp anchor join joinGrp link linkGrp timeline

msdescription: msDesc

namesdates: listNym listOrg listPerson listPlace

nets: eTree forest forestGrp graph tree

spoken: incident kinesic pause shift u vocal writing

tagdocs: classSpec eg egXML elementSpec listRef macroSpec moduleRef moduleSpec schemaSpec specGrp specGrpRef

textcrit: listWit witDetail

textstructure: argument byline dateline docAuthor docDate epigraph floatingText opener salute

transcr: addSpan damageSpan delSpan fw space

Declaration

```
element castList
{
   att.global.attributes,
   (
      ( model.divTop | model.global )*,
      ( ( model.common ), model.global* )*,
      ( ( castItem | castGroup ), model.global* )+,
      ( ( model.common ), model.global* )*
   )
}
```

Example

```
<castList>
  <castGroup>
    <head  rend="braced">Mendicants</head>
    <castItem>
      <role>Aafaa</role>
      <actor>Femi Johnson</actor>
    </castItem>
    <castItem>
      <role>Blindman</role>
      <actor>Femi Osofisan</actor>
    </castItem>
    <castItem>
      <role>Goyi</role>
      <actor>Wale Ogunyemi</actor>
    </castItem>
```

```
      <castItem>
        <role>Cripple</role>
        <actor>Tunji Oyelana</actor>
      </castItem>
    </castGroup>
    <castItem>
      <role>Si Bero</role>
      <roleDesc>Sister to Dr Bero</roleDesc>
      <actor>Deolo Adedoyin</actor>
    </castItem>
    <castGroup>
      <head  rend="braced">Two old women</head>
      <castItem>
        <role>Iya Agba</role>
        <actor>Nguba Agolia</actor>
      </castItem>
      <castItem>
        <role>Iya Mate</role>
        <actor>Bopo George</actor>
      </castItem>
    </castGroup>
    <castItem>
      <role>Dr Bero</role>
      <roleDesc>Specialist</roleDesc>
      <actor>Nat Okoro</actor>
    </castItem>
    <castItem>
      <role>Priest</role>
      <actor>Gbenga Sonuga</actor>
    </castItem>
    <castItem>
      <role>The old man</role>
      <roleDesc>Bero's father</roleDesc>
      <actor>Dapo Adelugba</actor>
    </castItem>
  </castList>
  <stage  type="mix">The action takes place in and around the home surgery of
  Dr Bero, lately returned from the wars.</stage>
```

<catDesc> (category description) describes some category within a taxonomy or text typology, either in the form of a brief prose description or in terms of the situational parameters used by the TEI formal textDesc.

Module header — *2. The TEI Header*

Attributes Global attributes only

Used by category

May contain

core: abbr address choice date distinct email emph expan foreign gloss measure measureGrp mentioned name num ptr ref rs soCalled term time title

corpus: textDesc

dictionaries: lang

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

tagdocs: att code gi ident tag val

transcr: am ex handShift subst

Declaration

```
element catDesc
{
   att.global.attributes,
   ( text | model.limitedPhrase | model.catDescPart )*
}
```

Example

```
<catDesc>Prose reportage</catDesc>
```

Example

```
<catDesc>
  <textDesc n="novel">
    <channel mode="w">print; part issues</channel>
    <constitution type="single"/>
    <derivation type="original"/>
    <domain type="art"/>
    <factuality type="fiction"/>
    <interaction type="none"/>
    <preparedness type="prepared"/>
    <purpose type="entertain" degree="high"/>
    <purpose type="inform" degree="medium"/>
  </textDesc>
</catDesc>
```

<catRef/> (category reference) specifies one or more defined categories within some taxonomy or text typology.

Module header — *2. The TEI Header*

Attributes In addition to global attributes

@target identifies the categories concerned

Status **Required**

Datatype 1–∞ occurrences of `data.pointer` separated by whitespace

Values A series of one or more space-separated pointers (URIs) to <category> elements, typically located within a <taxonomy> element inside a TEI header

@scheme identifies the classification scheme within which the set of categories concerned is defined

Status Optional

Datatype `data.pointer`

Values May supply the identifier of the associated <taxonomy> element.

Used by textClass

May contain Empty element

Declaration

```
element catRef
{
   att.global.attributes,
   attribute target { list { data.pointer+ } },
   attribute scheme { data.pointer }?,
   empty
}
```

Example

```
<catRef  target="#news #prov #sales2"/>
<!-- elsewhere -->
<taxonomy>
  <category  xml:id="news">
    <catDesc>Newspapers</catDesc>
  </category>
  <category  xml:id="prov">
    <catDesc>Provincial</catDesc>
  </category>
  <category  xml:id="sales2">
    <catDesc>Low to average annual sales</catDesc>
  </category>
</taxonomy>
```

Note The scheme attribute need be supplied only if more than one taxonomy has been declared

---

<catchwords> describes the system used to ensure correct ordering of the quires making up a codex or incunable, typically by means of annotations at the foot of the page.

Module msdescription — *10. Manuscript Description*

Attributes Global attributes only

Used by model.pPart.msdesc

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address binaryObject cb choice corr date del distinct email emph expan foreign gap gloss graphic hi index lb measure measureGrp mentioned milestone name note num orig pb ptr ref reg rs sic soCalled term time title unclear

dictionaries: lang oRef oVar pRef pVar

figures: formula

gaiji: g

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident specDesc specList tag val

textcrit: app witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element catchwords { att.global.attributes, macro.phraseSeq }
```

<category> contains an individual descriptive category, possibly nested within a superordinate category, within a user-defined taxonomy.

Module header — *2. The TEI Header*

Attributes Global attributes only

Used by category taxonomy

May contain

core: desc gloss

header: catDesc category

tagdocs: altIdent equiv

Declaration

```
element category
{
   att.global.attributes,
   ( ( catDesc | model.glossLike* ), category* )
}
```

Example

```
<category xml:id="b1">
  <catDesc>Prose reportage</catDesc>
</category>
```

Example

```
<category xml:id="b2">
  <catDesc>Prose
  </catDesc>
  <category xml:id="b11">
    <catDesc>reportage</catDesc>
  </category>
  <category xml:id="b12">
    <catDesc>fiction</catDesc>
  </category>
</category>
```

<cb/> (column break) marks the boundary between one column of a text and the next in a standard reference system.

Module core — *3. Elements Available in All TEI Documents*

Attributes In addition to global attributes

@ed (edition) indicates the edition or version in which the column break is located at this point

Status Recommended

Datatype `data.code`

Values Any string of characters; usually a siglum conventionally used for the edition.

Used by model.milestoneLike

May contain Empty element

Declaration

```
element cb { att.global.attributes, attribute ed { data.code }?, empty }
```

Example Markup of an early English dictionary printed in two columns:

```
<pb/>
<cb  n="1"/>
<entryFree>
  <form>Well</form>, <sense>a Pit to hold Spring-Water</sense>:
<sense>In the Art of <hi  rend="italic">War</hi>, a Depth the Miner
    sinks into the Ground, to find out and disappoint the Enemies Mines,
    or to prepare one</sense>.
</entryFree>
<entryFree>To <form>Welter</form>, <sense>to wallow</sense>, or
<sense>lie groveling</sense>.</entryFree>
<!-- remainder of column -->
<cb  n="2"/>
<entryFree>
  <form>Wey</form>, <sense>the greatest Measure for dry Things,
    containing five Chaldron</sense>.
</entryFree>
<entryFree>
  <form>Whale</form>, <sense>the greatest of
    Sea-Fishes</sense>.
</entryFree>
```

Source: [109]

Note On this element, the global **n** attribute indicates the number or other value associated with the column which follows the point of insertion of this <cb> element. Encoders should adopt a clear and consistent policy as to whether the numbers associated with column breaks relate to the physical sequence number of the column in the whole text, or whether columns are numbered within the page. By convention, the <cb> element is placed at the head of the column to which it refers.

<cell> contains one cell of a table.

Module figures — *14. Tables, Formulæ, and Graphics*

Attributes att.tableDecoration (*@role*, *@rows*, *@cols*)

Used by row

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address bibl biblStruct binaryObject cb choice cit corr date del desc distinct email emph expan foreign gap gloss graphic hi index label lb list listBibl measure measureGrp mentioned milestone name note num orig pb ptr q quote ref reg rs said sic soCalled stage term time title unclear

dictionaries: lang oRef oVar pRef pVar

drama: camera caption castList move sound tech view

figures: figure formula table

gaiji: g

header: biblFull

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material msDesc origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName listNym listOrg listPerson listPlace nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att classSpec code eg egXML elementSpec gi ident listRef macroSpec moduleRef moduleSpec specDesc specGrp specGrpRef specList tag val

textcrit: app listWit witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element cell
{
   att.global.attributes,
   att.tableDecoration.attributes,
   macro.paraContent}
```

Example

```
<row>
  <cell  role="label">General conduct</cell>
  <cell  role="data">Not satisfactory, on account of his great unpunctuality
     and inattention to duties</cell>
</row>
```

<certainty> indicates the degree of certainty or uncertainty associated with some aspect of the text markup.

Module certainty — *21. Certainty and Responsibility*

Attributes In addition to global attributes

@target points at the elements whose markup is uncertain.

Status **Required**

Datatype 1–∞ occurrences of `data.pointer` separated by whitespace

Values a series of one or more identifiers (URIs), separated by whitespace

```
Elizabeth went to <persName
   xml:id="ESX">Essex</persName>
<certainty
   target="#ESX"
   locus="gi"
   degree="0.6"/>
```

Note If more than one identifier is given, the <certainty> element is interpreted as applying to all. If no identifier is present on the element being annotated, the attribute should give the identifier of a <ptr> element which points at the element being annotated; for further discussion of this indirect pointing mechanism, see chapter *16. Linking, Segmentation, and Alignment*.

@locus indicates the precise location of the uncertainty in the markup: applicability of the element, precise position of the start- or end-tag, value of a specific attribute, etc.

Status **Required**

Datatype `data.enumerated`

Suggested values include: **gi** (element name) uncertain whether the element used actually applies to the passage.

startLoc (start location) start-tag may not be correctly located.

endLoc (end location) end-tag may not be correctly located.

location both the start-tag and the end-tag may not be correctly located.

attrName (attribute name) the value given for the attribute name is uncertain.

transcribedContent the content of the element may not be a correct transcription of the source text.

suppliedContent the content of the element may not have been correctly supplied by the reader, e.g. as in the cases of corr and abbrev elements.

Note If the name of an attribute is supplied, it must be prefixed by `att.`.

@assertedValue provides an alternative value for the aspect of the markup in question—an alternative generic identifier, transcription, or attribute value, or the identifier of an <anchor> element (to indicate an alternative starting or ending location). If an **assertedValue** is given, the confidence level specified by **degree** applies to the alternative markup specified by **assertedValue**; if none is given, it applies to the markup in the text.

Status Recommended

Datatype

```
data.pointer | data.name | data.word
```

Values generic identifier, attribute value, location (e.g. indicated by a reference to an <anchor> element or to an <ptr> element), or other appropriate alternative value.

```
<certainty
   target="#ESX"
   locus="gi"
   assertedValue="place"
   degree="0.2"/>
```

Note This attribute makes it possible to indicate the degree of confidence in a specific alternative to some aspect of the markup. In the example above the encoder is expressing the likelihood (.2) that the generic identifier should be <place> rather than <persName>, which is the coded element.

@given indicates conditions assumed in the assignment of a degree of confidence.

Status Recommended

Datatype 1–∞ occurrences of `data.pointer` separated by whitespace

Values a pointer to a characterization of the conditions which are assumed in the assignment of a degree of confidence.

Note A project may wish to control the vocabulary used in this attribute.The envisioned typical value of this attribute would be the identifier of another <certainty> element or a list of such identifiers. It may thus be possible to construct probability networks by chaining <certainty> elements together. Such networks would ultimately be grounded in unconditional <certainty> elements (with no value for **given**). The semantics of this chaining would be understood in this way: if a <certainty> element is specified, via a reference, as the assumption, then it is not the attribution of uncertainty that is the assumption, but rather the assertion itself. For instance, in the example above, the first <certainty> element indicates that the confidence in the identification of the new scribe as *msm*. The second indicates the degree of confidence that *Essex* is a personal name, given that the new scribe is *msm*. Note that the given in the second <certainty> element is not the assertion that the likelihood that msm is the new scribe is 0.6, but simply the assertion that msm is the new scribe; this is a recommended convention to facilitate building networks.The ambitious encoder may wish to attempt complex networks or probability assertions, experimenting with references to other elements or prose assertions, and deploying feature structure connectives such as <alt>, <join>, and <note>. However, we do not believe that the <certainty> element gives, at this time, a comprehensive ambiguity-free system for indicating certainty.

@degree indicates the degree of confidence assigned to the aspect of the markup named by the **locus** attribute.

Status Optional

Datatype `data.probability`

Used by model.global.meta

May contain

core: desc gloss

tagdocs: altIdent equiv

Declaration

```
element certainty
{
   att.global.attributes,
   attribute target { list { data.pointer+ } },
   attribute locus
   {
      "gi"
    | "startLoc"
    | "endLoc"
    | "location"
    | "attrName"
    | "transcribedContent"
    | "suppliedContent"
    | xsd:Name
   },
   attribute assertedValue { data.pointer | data.name | data.word }?,
   attribute given { list { data.pointer+ } }?,
   attribute degree { data.probability }?,
   model.glossLike*
}
```

Example (For discussion of this example, see section *21.1.2. Structured Indications of Uncertainty*)

```
Earnest went to <anchor  xml:id="A1"/> old
<persName  xml:id="SYB">Saybrook</persName>.

<certainty
   xml:id="c1"
   target="#SYB"
   locus="gi"
   degree="0.6"/>
<certainty
   target="#SYB"
   locus="startLoc"
   given="#c1"
   degree="0.9"/>
<certainty
   xml:id="C-c2"
   target="#SYB"
   locus="gi"
   assertedValue="persName"
   degree="0.4"/>
<certainty
   target="#SYB"
   locus="startLoc"
   given="#C-c2"
   degree="0.5"/>
<certainty
   target="#SYB"
   locus="startLoc"
   assertedValue="#a1"
   given="#c1"
   degree="0.5"/>
```

<change> summarizes a particular change or correction made to a particular version of an electronic text which is shared between several researchers.

Module header — *2. The TEI Header*

Attributes att.ascribed (*@who*)

@when supplies the date of the change in standard form, i.e. YYYY-MM-DD.

Status Mandatory when applicable

Datatype `data.temporal.w3c`

Values a date, time, or date & time in any of the formats defined in *XML Schema Part 2: Datatypes Second Edition*

Used by recordHist revisionDesc

May contain

analysis: interp interpGrp span spanGrp

certainty: certainty respons

core: abbr address bibl biblStruct cb choice cit date desc distinct email emph expan foreign gap gloss index label lb list listBibl measure measureGrp mentioned milestone name note num pb ptr q quote ref rs said soCalled stage term time title

dictionaries: lang

drama: camera caption castList move sound tech view

figures: figure table

header: biblFull

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp timeline

msdescription: catchwords depth dimensions height heraldry locus material msDesc origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName listNym listOrg listPerson listPlace nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att classSpec code eg egXML elementSpec gi ident listRef macroSpec moduleRef moduleSpec specGrp specGrpRef tag val

textcrit: listWit witDetail

transcr: addSpan am damageSpan delSpan ex fw handShift space subst

Declaration

```
element change
{
   att.global.attributes,
   att.ascribed.attributes,
   attribute when { data.temporal.w3c }?,
   ( text | model.limitedPhrase | model.inter | model.global )*
}
```

Example

```
<titleStmt>
  <title> ... </title>
```

```
  <editor xml:id="LDB">Lou Burnard</editor>
  <respStmt xml:id="BZ">
    <resp>copy editing</resp>
    <name>Brett Zamir</name>
  </respStmt>
</titleStmt>
<!-- ... -->
<revisionDesc>
  <change who="#BZ" when="2008-02-02">Finished chapter 23</change>
  <change who="#BZ" when="2008-01-02">Finished chapter 2</change>
  <change n="P2.2" when="1991-12-21" who="#LDB">Added examples to section 3</change>
  <change when="1991-11-11" who="#MSM">Deleted chapter 10</change>
</revisionDesc>
```

Note The **who** attribute may be used to point to any other element, but will typically specify a <respStmt> or <person> element elsewhere in the header, identifying the person responsible for the change and their role in making it.It is recommended that changes be recorded with the most recent first.

<channel> (primary channel) describes the medium or channel by which a text is delivered or experienced. For a written text, this might be print, manuscript, e-mail, etc.; for a spoken one, radio, telephone, face-to-face, etc.

Module corpus — *15. Language Corpora*

Attributes In addition to global attributes

@mode specifies the mode of this channel with respect to speech and writing.

Status Optional

Legal values are: **s** (spoken)

w (written)

sw (spoken to be written) e.g. dictation

ws (written to be spoken) e.g. a script

m (mixed)

x (unknown or inapplicable) **[Default]**

Used by model.textDescPart

May contain

analysis: interp interpGrp span spanGrp

certainty: certainty respons

core: abbr address cb choice date distinct email emph expan foreign gap gloss index lb measure measureGrp mentioned milestone name note num pb ptr ref rs soCalled term time title

dictionaries: lang

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat

geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident tag val

textcrit: witDetail

transcr: addSpan am damageSpan delSpan ex fw handShift space subst

Declaration

```
element channel
{
   att.global.attributes,
   attribute mode { "s" | "w" | "sw" | "ws" | "m" | "x" }?,
   macro.phraseSeq.limited}
```

Example

```
<channel  mode="s">face-to-face conversation</channel>
```

<char> (character) provides descriptive information about a character.

Module gaiji — *5. Representation of Non-standard Characters and Glyphs*

Attributes Global attributes only

Used by charDecl

May contain

core: binaryObject desc gloss graphic note

figures: formula

gaiji: charName charProp mapping

tagdocs: altIdent equiv

textcrit: witDetail

Declaration

```
element char
{
   att.global.attributes,
   (
      charName?,
      model.glossLike*,
      charProp*,
      mapping*,
      model.graphicLike*,
      model.noteLike*
   )
}
```

Example

```
<char  xml:id="circledU4EBA">
  <charName>CIRCLED IDEOGRAPH 4EBA</charName>
  <charProp>
```

```
    <unicodeName>character-decomposition-mapping</unicodeName>
    <value>circle</value>
  </charProp>
  <charProp>
    <localName>daikanwa</localName>
    <value>36</value>
  </charProp>
  <mapping type="standard">人
  </mapping>
</char>
```

<charDecl> (character declarations) provides information about nonstandard characters and glyphs.

Module gaiji — *5. Representation of Non-standard Characters and Glyphs*

Attributes Global attributes only

Used by model.encodingPart

May contain

core: desc

gaiji: char glyph

Declaration

```
element charDecl { att.global.attributes, ( desc?, ( char | glyph )+ ) }
```

Example

```
<charDecl>
  <char xml:id="aENL">
    <charName>LATIN LETTER ENLARGED SMALL A</charName>
    <mapping type="standardized">a</mapping>
  </char>
</charDecl>
```

<charName> (character name) contains the name of a character, expressed following Unicode conventions.

Module gaiji — *5. Representation of Non-standard Characters and Glyphs*

Attributes Global attributes only

Used by char

May contain Character data only

Declaration `element charName { att.global.attributes, text }`

Example

```
<charName>CIRCLED IDEOGRAPH 4EBA</charName>
```

Note The name must follow Unicode conventions for character naming. Projects working in similar fields are recommended to coordinate and publish their list of <charName>s to facilitate data exchange.

<charProp> (character property) provides a name and value for some property of the parent character or glyph.

Module gaiji — *5. Representation of Non-standard Characters and Glyphs*

Attributes att.typed (*@type*, *@subtype*)

Used by char glyph

May contain

gaiji: localName unicodeName value

Declaration

```
element charProp
{
   att.global.attributes,
   att.typed.attributes,
   ( ( unicodeName | localName ), value )
}
```

Example

```
<charProp>
  <unicodeName>character-decomposition-mapping</unicodeName>
  <value>circle</value>
</charProp>
<charProp>
  <localName>daikanwa</localName>
  <value>36</value>
</charProp>
```

Note If the property is a Unicode Normative Property, then its <unicodeName> must be supplied. Otherwise, its name must be specied by means of a <localName>.At a later release, additional constraints will be defined on possible value/name combinations using Schematron rules

<choice> groups a number of alternative encodings for the same point in a text.

Module core — *3. Elements Available in All TEI Documents*

Attributes Global attributes only

Used by choice model.pPart.editorial

May contain

core: abbr choice corr expan orig reg sic unclear

linking: seg

transcr: am ex

Declaration

```
element choice { att.global.attributes, ( model.choicePart | choice )* }
```

Example An American encoding of *Gulliver's Travels* which retains the British spelling but also provides a version regularized to American spelling might be encoded as follows.

```
<p>Lastly, That, upon his solemn oath to observe all the above
articles, the said man-mountain shall have a daily allowance of
meat and drink sufficient for the support of <choice>
   <sic>1724</sic>
   <corr>1728</corr>
  </choice> of our subjects,
with free access to our royal person, and other marks of our
<choice>
   <orig>favour</orig>
   <reg>favor</reg>
  </choice>.</p>
```

Source: [188]

Note Because the children of a <choice> element all represent alternative ways of encoding the same sequence, it is natural to think of them as mutually exclusive. However, there may be cases where a full representation of a text requires the alternative encodings to be considered as parallel. Note also that <choice> elements may self-nest.For a specialized version of <choice> for encoding multiple witnesses of a single work, see section *12.1. The Apparatus Entry, Readings, and Witnesses*.

<cit> (cited quotation) contains a quotation from some other document, together with a bibliographic reference to its source. In a dictionary it may contain an example text with at least one occurrence of the word form, used in the sense being described, or a translation of the headword, or an example.

Module core — *3. Elements Available in All TEI Documents*

Attributes att.typed (*@type*, *@subtype*)

Used by model.quoteLike model.entryPart.top

May contain

analysis: interp interpGrp span spanGrp

certainty: certainty respons

core: bibl biblStruct cb cit gap index lb milestone note pb ptr q quote ref said

dictionaries: case colloc def etym form gen gramGrp hom hyph iType lbl mood number orth per pos pron re sense stress subc superEntry syll tns usg xr

header: biblFull

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp timeline

msdescription: msDesc

spoken: incident kinesic pause shift vocal writing

textcrit: witDetail

transcr: addSpan damageSpan delSpan fw space

Declaration

```
element cit
{
   att.global.attributes,
   att.typed.attributes,
   (
```

```
      model.qLike    | model.biblLike    | model.ptrLike    | model.global    | model.entryPart   )+
}
```

Example

```
<cit>
  <quote>and the breath of the whale is frequently attended with such an insupportable smell,
     as to bring on disorder of the brain.</quote>
  <bibl>Ulloa's South America</bibl>
</cit>
```

Source: [140]

Example

```
<entry>
  <form>
    <orth>horrifier</orth>
  </form>
  <cit  type="translation"  xml:lang="en">
    <quote>to horrify</quote>
  </cit>
  <cit  type="example">
    <quote>elle était horrifiée par la dépense</quote>
    <cit  type="translation"  xml:lang="en">
      <quote>she was horrified at the expense.</quote>
    </cit>
  </cit>
</entry>
```

<cl> (clause) represents a grammatical clause.

Module analysis — *17. Simple Analytic Mechanisms*

Attributes att.segLike (*@type*, *@function*, *@part*) att.metrical (*@met*, *@real*, *@rhyme*)

Used by model.segLike

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address binaryObject cb choice corr date del distinct email emph expan foreign gap gloss graphic hi index lb measure measureGrp mentioned milestone name note num orig pb ptr ref reg rs sic soCalled term time title unclear

dictionaries: lang oRef oVar pRef pVar

figures: formula

gaiji: g

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat

geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident specDesc specList tag val

textcrit: app witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element cl
{
   att.global.attributes,
   att.segLike.attributes,
   att.metrical.attributes,
   macro.phraseSeq}
```

Example

```
<cl type="relative" function="clause_modifier">Which frightened
both the heroes so,<cl>They quite forgot their quarrel.</cl>
</cl>
```

<classCode>

<classCode> (classification code) contains the classification code used for this text in some standard classification system.

Module header — *2. The TEI Header*

Attributes In addition to global attributes

@scheme identifies the classification system or taxonomy in use.

Status **Required**

Datatype `data.pointer`

Values may point to a local definition, for example in a <taxonomy> element, or more usually to some external location where the scheme is fully defined.

Used by textClass

May contain

analysis: interp interpGrp span spanGrp

certainty: certainty respons

core: abbr address cb choice date distinct email emph expan foreign gap gloss index lb measure measureGrp mentioned milestone name note num pb ptr ref rs soCalled term time title

dictionaries: lang

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident tag val

textcrit: witDetail

transcr: addSpan am damageSpan delSpan ex fw handShift space subst

Declaration

```
element classCode
{
   att.global.attributes,
   attribute scheme { data.pointer },
   macro.phraseSeq.limited}
```

Example

```
<classCode scheme="http://www.udc.org">410</classCode>
```

<classDecl>

(classification declarations) contains one or more taxonomies defining any classificatory codes used elsewhere in the text.

Module header — *2. The TEI Header*

Attributes Global attributes only

Used by model.encodingPart

May contain

header: taxonomy

Declaration

```
element classDecl { att.global.attributes, taxonomy+ }
```

Example

```
<classDecl>
  <taxonomy xml:id="LCSH">
    <bibl>Library of Congress Subject Headings</bibl>
  </taxonomy>
</classDecl>
<!-- ... -->
<textClass>
  <keywords scheme="#LCSH">
    <list>
      <item>Political science</item>
      <item>United States -- Politics and government —
            Revolution, 1775-1783</item>
    </list>
  </keywords>
</textClass>
```

<classSpec> (class specification) contains reference information for a TEI element class; that is a group of elements which appear together in content models, or which share some common attribute, or both.

Module tagdocs — *22. Documentation Elements*

Attributes att.identified (*@ident*, *@predeclare*, *@module*, *@mode*)

@type indicates whether this is a model class or an attribute class

Status **Required**

Legal values are: **model** (content model) members of this class appear in the same content models

atts (attributes) members of this class share common attributes

@generate indicates which alternation and sequence instantiations of a model class may be referenced. By default, all variations are permitted.

Status Optional

Datatype 1–5 occurrences of `data.enumerated` separated by whitespace

Legal values are: **alternation** members of the class are alternatives

sequence members of the class are to be provided in sequence

sequenceOptional members of the class may be provided, in sequence, but are optional

sequenceOptionalRepeatable members of the class may be provided one or more times, in sequence, but are optional.

sequenceRepeatable members of the class may be provided one or more times, in sequence

Used by model.oddDecl

May contain

core: desc gloss

tagdocs: altIdent attList classes equiv exemplum listRef remarks

Declaration

```
element classSpec
{
   att.global.attributes,
   att.identified.attributes,
   attribute type { "model" | "atts" },
   attribute generate
   {
      list
      {
         (
            "alternation"
          | "sequence"
          | "sequenceOptional"
          | "sequenceOptionalRepeatable"
          | "sequenceRepeatable"
         ),
         (
            "alternation"
          | "sequence"
          | "sequenceOptional"
          | "sequenceOptionalRepeatable"
          | "sequenceRepeatable"
```

```
         )*
      }
   }?,
   ( model.glossLike*, classes?, attList?, exemplum*, remarks*, listRef* )
}
```

Example

```
<classSpec module="tei" type="model" ident="model.segLike">
  <desc>groups elements used for arbitrary segmentation. </desc>
  <classes>
    <memberOf key="model.phrase"/>
  </classes>
  <remarks>
    <p>The principles on which segmentation is carried out, and
        any special codes or attribute values used, should be defined explicitly
        in the <gi>segmentation</gi> element of the <gi>encodingDesc</gi> within
        the associated TEI header.</p>
  </remarks>
</classSpec>
```

<classes> specifies all the classes of which the documented element or class is a member or subclass.

Module tagdocs — *22. Documentation Elements*

Attributes In addition to global attributes

@mode specifies the effect of this declaration on its parent module.

Status Optional

Legal values are: **change** this declaration changes the declaration of the same name in the current definition

replace this declaration replaces the declaration of the same name in the current definition **[Default]**

Used by classSpec elementSpec

May contain

tagdocs: memberOf

Declaration

```
element classes
{
   att.global.attributes,
   attribute mode { "change" | "replace" }?,
   ( memberOf* )
}
```

Example

```
<classes>
  <memberOf key="model.qLike"/>
  <memberOf key="att.declarable"/>
</classes>
```

This <classes> element indicates that the element documented (which may be an element or a class) is a member of two distinct classes: model.qLike and att.declarable.

Note An empty <classes> element indicates that the element documented is not a member of any class. This should not generally happen.

<climate> contains information about the physical climate of a place.

Module namesdates — *13. Names, Dates, People, and Places*

Attributes att.datableatt.datable.w3c (*@period*, *@when*, *@notBefore*, *@notAfter*, *@from*, *@to*) att.datable.iso (*@when-iso*, *@notBefore-iso*, *@notAfter-iso*, *@from-iso*, *@to-iso*) att.editLike (*@cert*, *@resp*, *@evidence*, *@source*, *@precision*) att.naming (*@key*, *@ref*, *@nymRef*) att.typed (*@type*, *@subtype*)

Used by climate model.placeTraitLike

May contain

core: bibl biblStruct desc head label note p

header: biblFull

linking: ab

msdescription: msDesc

namesdates: climate

textcrit: witDetail

Declaration

```
element climate
{
   att.global.attributes,
   att.datable.w3c.attributes,
   att.datable.iso.attributes,
   att.editLike.attributes,
   att.naming.attributes,
   att.typed.attributes,
   (
      model.headLike*,
      ( ( model.pLike+ ) | ( model.labelLike+ ) ),
      ( model.noteLike | model.biblLike )*,
      climate*
   )
}
```

Example

```
<place xml:id="ROMA">
  <placeName>Rome</placeName>
<!-- ... -->
  <climate>
    <ab>
      <table>
        <head>24-hr Average Temperature</head>
        <row>
          <cell/>
          <cell role="label">Jan</cell>
          <cell role="label">Jun</cell>
          <cell role="label">Dec</cell>
        </row>
```

```
          <row>
            <cell role="label">°C</cell>
            <cell role="data">7.1</cell>
            <cell role="data">21.7</cell>
            <cell role="data">8.3</cell>
          </row>
          <row>
            <cell role="label">°F</cell>
            <cell role="data">44.8</cell>
            <cell role="data">71.1</cell>
            <cell role="data">46.9</cell>
          </row>
        </table>
      </ab>
      <note>Taken from <bibl>
          <abbr>GHCN 2 Beta</abbr>: The Global
              Historical Climatology Network, version 2 beta, 1904 months between
              1811 and 1980. <ptr
             target="http://www.worldclimate.com/cgi-bin/data.pl?ref=N41E012+1202+0004058G2"/>
        </bibl>
      </note>
    </climate>
  </place>
```

<closer> groups together salutations, datelines, and similar phrases appearing as a final group at the end of a division, especially of a letter.

Module textstructure — *4. Default Text Structure*

Attributes Global attributes only

Used by model.divBottomPart

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address binaryObject cb choice corr date del distinct email emph expan foreign gap gloss graphic hi index lb measure measureGrp mentioned milestone name note num orig pb ptr ref reg rs sic soCalled term time title unclear

dictionaries: lang oRef oVar pRef pVar

figures: formula

gaiji: g

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident specDesc specList tag val

textcrit: app witDetail

textstructure: dateline salute signed

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element closer
{
   att.global.attributes,
   (
      text
    | model.gLike    | signed    | dateline    | salute    | model.phrase    | model.global   )*
}
```

Example

```
<div type="letter">
  <p>perhaps you will favour me with a sight
     of it when convenient.</p>
  <closer>
    <salute>I remain, &c. &c.</salute>
    <signed>H. Colburn</signed>
  </closer>
</div>
```

Example

```
<div type="chapter">
  <p>and his heart was going like mad and yes
     I said yes I will Yes.</p>
  <closer>
    <dateline>
      <name type="place">Trieste-Zürich-Paris,</name>
      <date>1914–1921</date>
    </dateline>
  </closer>
</div>
```

<code> contains literal code from some formal language such as a programming language.

Module tagdocs — *22. Documentation Elements*

Attributes In addition to global attributes

@lang (formal language) a name identifying the formal language in which the code is expressed

Status Optional

Datatype `data.word`

Used by model.emphLike

May contain Character data only

Declaration

```
element code { att.global.attributes, attribute lang { data.word }?, text }
```

Example

```
<code  lang="JAVA"> Size fCheckbox1Size = new Size();
fCheckbox1Size.Height = 500;
fCheckbox1Size.Width = 500;
xCheckbox1.setSize(fCheckbox1Size);
</code>
```

<collation> contains a description of how the leaves or bifolia are physically arranged.

Module msdescription — *10. Manuscript Description*

Attributes Global attributes only

Used by supportDesc

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address bibl biblStruct binaryObject cb choice cit corr date del desc distinct email emph expan foreign gap gloss graphic hi index l label lb lg list listBibl measure measureGrp mentioned milestone name note num orig p pb ptr q quote ref reg rs said sic soCalled sp stage term time title unclear

dictionaries: lang oRef oVar pRef pVar

drama: camera caption castList move sound tech view

figures: figure formula table

gaiji: g

header: biblFull

iso-fs: fLib fs fvLib

linking: ab alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material msDesc origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName listNym listOrg listPerson listPlace nameLink offset orgName persName placeName region roleName settlement state surname

nets: eTree forest forestGrp graph tree

spoken: incident kinesic pause shift u vocal writing

tagdocs: att classSpec code eg egXML elementSpec gi ident listRef macroSpec moduleRef moduleSpec schemaSpec specDesc specGrp specGrpRef specList tag val

textcrit: app listWit witDetail

textstructure: floatingText

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element collation { att.global.attributes, macro.specialPara }
```

Example

```
<collation>The written leaves preceded by an original flyleaf,
conjoint with the pastedown.</collation>
```

Example

```
<collation>
  <p>
    <formula>1-5.8 6.6 (catchword, f. 46, does not match following text)
        7-8.8 9.10, 11.2 (through f. 82) 12-14.8 15.8(-7)</formula>
    <catchwords>Catchwords are written horizontally in center
        or towards the right lower margin in various manners:
        in red ink for quires 1-6 (which are also signed in red
        ink with letters of the alphabet and arabic numerals);
        quires 7-9 in ink of text within yellow decorated frames;
        quire 10 in red decorated frame; quire 12 in ink of text;
        quire 13 with red decorative slashes; quire 14 added in
        cursive hand.</catchwords>
  </p>
</collation>
```

<collection> contains the name of a collection of manuscripts, not necessarily located within a single repository.

Module msdescription — *10. Manuscript Description*

Attributes att.naming (*@key*, *@ref*, *@nymRef*)

Used by altIdentifier msIdentifier

May contain

gaiji: g

Declaration

```
element collection
{
   att.global.attributes,
   att.naming.attributes,
   macro.xtext}
```

Example

```
<msIdentifier>
  <country>USA</country>
  <region>California</region>
  <settlement>San Marino</settlement>
  <repository>Huntington Library</repository>
  <collection>Ellesmere</collection>
  <idno>El 26 C 9</idno>
  <msName>The Ellesmere Chaucer</msName>
</msIdentifier>
```

<colloc> (collocate) contains a collocate of the headword.

Module dictionaries — *9. Dictionaries*

Attributes att.lexicographic (*@expand*, *@norm*, *@split*, *@value*, *@orig*, *@location*, *@mergedIn*, *@opt*)
att.typed (*@type*, *@subtype*)

Used by model.entryPart model.gramPart

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address bibl biblStruct binaryObject cb choice cit corr date del desc distinct email emph expan foreign gap gloss graphic hi index label lb list listBibl measure measureGrp mentioned milestone name note num orig pb ptr q quote ref reg rs said sic soCalled stage term time title unclear

dictionaries: lang oRef oVar pRef pVar

drama: camera caption castList move sound tech view

figures: figure formula table

gaiji: g

header: biblFull

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material msDesc origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName listNym listOrg listPerson listPlace nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att classSpec code eg egXML elementSpec gi ident listRef macroSpec moduleRef moduleSpec specDesc specGrp specGrpRef specList tag val

textcrit: app listWit witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element colloc
{
   att.global.attributes,
   att.lexicographic.attributes,
   att.typed.attributes,
   macro.paraContent}
```

Example

```
<entry>
  <form>
    <orth>médire</orth>
  </form>
  <gramGrp>
    <colloc  type="prep">de</colloc>
```

```
  </gramGrp>
</entry>
```

<colophon> contains the colophon of a manuscript item: that is, a statement providing information regarding the date, place, agency, or reason for production of the manuscript.

Module msdescription — *10. Manuscript Description*

Attributes Global attributes only

Used by msItemStruct model.msItemPart

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address binaryObject cb choice corr date del distinct email emph expan foreign gap gloss graphic hi index lb measure measureGrp mentioned milestone name note num orig pb ptr ref reg rs sic soCalled term time title unclear

dictionaries: lang oRef oVar pRef pVar

figures: formula

gaiji: g

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident specDesc specList tag val

textcrit: app witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element colophon { att.global.attributes, macro.phraseSeq }
```

Example

```
<colophon>Ricardus Franciscus Scripsit Anno Domini
1447.</colophon>
```

Example

```
<colophon>Explicit expliceat/scriptor ludere eat.</colophon>
```

Example

```
<colophon>Explicit venenum viciorum domini illius, qui comparavit Anno
domini Millessimo Trecentesimo nonagesimo primo, Sabbato in festo
sancte Marthe virginis gloriose. Laus tibi criste quia finitur
libellus iste.</colophon>
```

<cond> (conditional feature-structure constraint) defines a conditional feature-structure constraint; the consequent and the antecedent are specified as feature structures or feature-structure collections; the constraint is satisfied if both the antecedent and the consequent subsume a given feature structure, or if the antecedent does not.

Module iso-fs — *18. Feature Structures*

Attributes Global attributes only

Used by fsConstraints

May contain

iso-fs: f fs then

Declaration

```
element cond { att.global.attributes, ( ( fs | f ), then, ( fs | f ) ) }
```

Example

```
<cond>
  <fs>
    <f name="BAR">
      <symbol value="1"/>
    </f>
  </fs>
  <then/>
  <fs>
    <f name="SUBCAT">
      <binary value="false"/>
    </f>
  </fs>
</cond>
```

Note May contain an antecedent feature structure, an empty <then> element, and a consequent feature structure.

<condition> contains a description of the physical condition of the manuscript.

Module msdescription — *10. Manuscript Description*

Attributes Global attributes only

Used by supportDesc

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address bibl biblStruct binaryObject cb choice cit corr date del desc distinct email emph expan foreign gap gloss graphic hi index l label lb lg list listBibl measure

measureGrp mentioned milestone name note num orig p pb ptr q quote ref reg rs said sic soCalled sp stage term time title unclear

dictionaries: lang oRef oVar pRef pVar

drama: camera caption castList move sound tech view

figures: figure formula table

gaiji: g

header: biblFull

iso-fs: fLib fs fvLib

linking: ab alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material msDesc origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName listNym listOrg listPerson listPlace nameLink offset orgName persName placeName region roleName settlement state surname

nets: eTree forest forestGrp graph tree

spoken: incident kinesic pause shift u vocal writing

tagdocs: att classSpec code eg egXML elementSpec gi ident listRef macroSpec moduleRef moduleSpec schemaSpec specDesc specGrp specGrpRef specList tag val

textcrit: app listWit witDetail

textstructure: floatingText

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element condition { att.global.attributes, macro.specialPara }
```

Example

```
<condition>
  <p>There are lacunae in three places in this
     manuscript. After 14v two
     leaves has been cut out and narrow strips leaves remains in the spine. After
     68v one gathering is missing and after 101v at least one gathering of 8 leaves
     has been lost. </p>
  <p>Several leaves are damaged with tears or holes or have a
     irregular shape. Some of the damages do not allow the lines to be of full
     length and they are apparently older than the script. There are tears on fol.
     2r-v, 9r-v, 10r-v, 15r-18v, 19r-v, 20r-22v, 23r-v, 24r-28v, 30r-v, 32r-35v,
     37r-v, 38r-v, 40r-43v, 45r-47v, 49r-v, 51r-v, 53r-60v, 67r-v, 68r-v, 70r-v,
     74r-80v, 82r-v, 86r-v, 88r-v, 89r-v, 95r-v, 97r-98v 99r-v, 100r-v. On fol. 98
     the corner has been torn off. Several leaves are in a bad condition due to
     moist and wear, and have become dark, bleached or
     wrinkled. </p>
  <p>The script has been
     touched up in the 17th century with black ink. The touching up on the following
     fols. was done by
  <name>Bishop Brynjólf Sveinsson</name>: 1v, 3r, 4r, 5r,
     6v, 8v,9r, 10r, 14r, 14v, 22r,30v, 36r-52v, 72v, 77r,78r,103r, 104r,. An
```

```
      AM-note says according to the lawman
    <name>Sigurður Björnsson</name> that the rest of the
      touching up was done by himself and another lawman
    <name>Sigurður Jónsson</name>.
    <name>Sigurður Björnsson</name> did the touching up
      on the following fols.: 46v, 47r, 48r, 49r-v, 50r, 52r-v.
    <name>Sigurður Jónsson</name> did the rest of the
      touching up in the section 36r-59r containing
    <title>Bretasögur</title>
    </p>
  </condition>
```

<constitution> describes the internal composition of a text or text sample, for example as fragmentary, complete, etc.

Module corpus — *15. Language Corpora*

Attributes In addition to global attributes

@type specifies how the text was constituted.

Status Optional

Legal values are: **single** a single complete text [**Default**]

composite a text made by combining several smaller items, each individually complete

frags (fragments) a text made by combining several smaller, not necessarily complete, items

unknown composition unknown or unspecified

Used by model.textDescPart

May contain

analysis: interp interpGrp span spanGrp

certainty: certainty respons

core: abbr address cb choice date distinct email emph expan foreign gap gloss index lb measure measureGrp mentioned milestone name note num pb ptr ref rs soCalled term time title

dictionaries: lang

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident tag val

textcrit: witDetail

transcr: addSpan am damageSpan delSpan ex fw handShift space subst

Declaration

```
element constitution
{
   att.global.attributes,
   attribute type { "single" | "composite" | "frags" | "unknown" }?,
   macro.phraseSeq.limited}
```

Example

```
<constitution type="frags">Prologues only.</constitution>
```

Note The function of this element seems to overlap with both the **org** attribute on and the <samplingDecl> in the <encodingDesc>.

<content> (content model) contains the text of a declaration for the schema documented.

Module tagdocs — *22. Documentation Elements*

Attributes Global attributes only

Used by elementSpec macroSpec moduleRef

May contain

tagdocs: valList

Declaration

```
element content { att.global.attributes, ( macro.schemaPattern | valList )* }
```

Example This content model allows either a sequence of paragraphs or a series of msItem elements optionally preceded by a summary:

```
<content>
  <rng:choice>
    <rng:oneOrMore>
      <rng:ref name="model.pLike"/>
    </rng:oneOrMore>
    <rng:group>
      <rng:optional>
        <rng:ref name="summary"/>
      </rng:optional>
      <rng:oneOrMore>
        <rng:ref name="msItem"/>
      </rng:oneOrMore>
    </rng:group>
  </rng:choice>
</content>
```

Note As the example shows, content models in P5 may be expressed using the RelaxNG syntax directly. More exactly, they are defined using the pattern `macro.schemaPattern`. Alternatively, a content model may be expressed using the TEI <valList> element.

<corr> (correction) contains the correct form of a passage apparently erroneous in the copy text.

Module core — *3. Elements Available in All TEI Documents*

Attributes att.editLike (*@cert*, *@resp*, *@evidence*, *@source*, *@precision*) att.typed (*@type*, *@subtype*)

Used by model.pPart.transcriptional model.choicePart

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address bibl biblStruct binaryObject cb choice cit corr date del desc distinct email emph expan foreign gap gloss graphic hi index label lb list listBibl measure measureGrp mentioned milestone name note num orig pb ptr q quote ref reg rs said sic soCalled stage term time title unclear

dictionaries: lang oRef oVar pRef pVar

drama: camera caption castList move sound tech view

figures: figure formula table

gaiji: g

header: biblFull

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material msDesc origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName listNym listOrg listPerson listPlace nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att classSpec code eg egXML elementSpec gi ident listRef macroSpec moduleRef moduleSpec specDesc specGrp specGrpRef specList tag val

textcrit: app listWit witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element corr
{
   att.global.attributes,
   att.editLike.attributes,
   att.typed.attributes,
   macro.paraContent}
```

Example If all that is desired is to call attention to the fact that the copy text has been corrected, <corr> may be used alone:

```
I don't know,
Juan. It's so far in the past now — how <corr>can we</corr> prove
or disprove anyone's theories?
```

Example It is also possible, using the <choice> and <sic> elements, to provide an uncorrected reading:

```
I don't know, Juan. It's so far in the past now —
how <choice>
  <sic>we can</sic>
```

```
  <corr>can we</corr>
</choice> prove or
disprove anyone's theories?
```

<correction> (correction principles) states how and under what circumstances corrections have been made in the text.

Module header — *2. The TEI Header*

Attributes att.declarable (*@default*)

@status indicates the degree of correction applied to the text.

Status Optional

Legal values are: **high** the text has been thoroughly checked and proofread.

medium the text has been checked at least once.

low the text has not been checked.

unknown the correction status of the text is unknown. [**Default**]

@method indicates the method adopted to indicate corrections within the text.

Status Optional

Legal values are: **silent** corrections have been made silently [**Default**]

markup corrections have been represented using markup

Used by model.editorialDeclPart

May contain

core: p

linking: ab

Declaration

```
element correction
{
   att.global.attributes,
   att.declarable.attributes,
   attribute status { "high" | "medium" | "low" | "unknown" }?,
   attribute method { "silent" | "markup" }?,
   model.pLike+
}
```

Example

```
<correction>
  <p>Errors in transcription controlled by using the
     WordPerfect spelling checker, with a user defined dictionary of 500
     extra words taken from Chambers Twentieth Century Dictionary.</p>
</correction>
```

Note May be used to note the results of proof reading the text against its original, indicating (for example) whether discrepancies have been silently rectified, or recorded using the editorial tags described in section *3.4. Simple Editorial Changes*.

<country> contains the name of a geo-political unit, such as a nation, country, colony, or commonwealth, larger than or administratively superior to a region and smaller than a bloc.

Module namesdates — *13. Names, Dates, People, and Places*

Attributes att.naming (*@key*, *@ref*, *@nymRef*) att.typed (*@type*, *@subtype*) att.datableatt.datable.w3c (*@period*, *@when*, *@notBefore*, *@notAfter*, *@from*, *@to*) att.datable.iso (*@when-iso*, *@notBefore-iso*, *@notAfter-iso*, *@from-iso*, *@to-iso*)

Used by model.placeNamePart

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address binaryObject cb choice corr date del distinct email emph expan foreign gap gloss graphic hi index lb measure measureGrp mentioned milestone name note num orig pb ptr ref reg rs sic soCalled term time title unclear

dictionaries: lang oRef oVar pRef pVar

figures: formula

gaiji: g

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident specDesc specList tag val

textcrit: app witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element country
{
   att.global.attributes,
   att.naming.attributes,
   att.typed.attributes,
   att.datable.w3c.attributes,
   att.datable.iso.attributes,
   macro.phraseSeq}
```

Example

```
<country key="DK">Denmark</country>
```

Note The recommended source for codes to represent coded country names is ISO 3166.

<creation> contains information about the creation of a text.

Module header — *2. The TEI Header*

Attributes Global attributes only

Used by profileDesc

May contain

analysis: interp interpGrp span spanGrp

certainty: certainty respons

core: abbr address cb choice date distinct email emph expan foreign gap gloss index lb measure measureGrp mentioned milestone name note num pb ptr ref rs soCalled term time title

dictionaries: lang

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident tag val

textcrit: witDetail

transcr: addSpan am damageSpan delSpan ex fw handShift space subst

Declaration

```
element creation { att.global.attributes, macro.phraseSeq.limited }
```

Example

```
<creation>
  <date>Before 1987</date>
</creation>
```

Example

```
<creation>
  <date  when="1988-07-10">10 July 1988</date>
</creation>
```

Note Character data and phrase-level elements.The <creation> element may be used to record details of a text's creation, e.g. the date and place it was composed, if these are of interest; it should not be confused with the <publicationStmt> element, which records date and place of publication.

<custEvent> (custodial event) describes a single event during the custodial history of a manuscript.

Module msdescription — *10. Manuscript Description*

Attributes att.datableatt.datable.w3c (*@period*, *@when*, *@notBefore*, *@notAfter*, *@from*, *@to*) att.datable.iso (*@when-iso*, *@notBefore-iso*, *@notAfter-iso*, *@from-iso*, *@to-iso*) att.typed (*@type*, *@subtype*)

Used by custodialHist

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address bibl biblStruct binaryObject cb choice cit corr date del desc distinct email emph expan foreign gap gloss graphic hi index l label lb lg list listBibl measure measureGrp mentioned milestone name note num orig p pb ptr q quote ref reg rs said sic soCalled sp stage term time title unclear

dictionaries: lang oRef oVar pRef pVar

drama: camera caption castList move sound tech view

figures: figure formula table

gaiji: g

header: biblFull

iso-fs: fLib fs fvLib

linking: ab alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material msDesc origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName listNym listOrg listPerson listPlace nameLink offset orgName persName placeName region roleName settlement state surname

nets: eTree forest forestGrp graph tree

spoken: incident kinesic pause shift u vocal writing

tagdocs: att classSpec code eg egXML elementSpec gi ident listRef macroSpec moduleRef moduleSpec schemaSpec specDesc specGrp specGrpRef specList tag val

textcrit: app listWit witDetail

textstructure: floatingText

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element custEvent
{
   att.global.attributes,
   att.datable.w3c.attributes,
   att.datable.iso.attributes,
   att.typed.attributes,
   macro.specialPara}
```

Example

```
<custEvent  type="photography">Photographed by David Cooper on <date>12 Dec 1964</date>
</custEvent>
```

<custodialHist> (custodial history) contains a description of a manuscript's custodial history, either as running prose or as a series of dated custodial events.

Module msdescription — *10. Manuscript Description*

Attributes Global attributes only

Used by adminInfo

May contain

core: p

linking: ab

msdescription: custEvent

Declaration

```
element custodialHist { att.global.attributes, ( model.pLike+ | custEvent+ ) }
```

Example

```
<custodialHist>
  <custEvent type="conservation" notBefore="1961-03" notAfter="1963-02">Conserved between March
1961 and February 1963 at
    Birgitte Dalls Konserveringsværksted.</custEvent>
  <custEvent type="photography" notBefore="1988-05-01" notAfter="1988-05-30">Photographed in
    May 1988 by AMI/FA.</custEvent>
  <custEvent type="transfer-dispatch" notBefore="1989-11-13" notAfter="1989-11-13">Dispatched to
Iceland
    13 November 1989.</custEvent>
</custodialHist>
```

<damage> contains an area of damage to the text witness.

Module transcr — *11. Representation of Primary Sources*

Attributes att.typed (*@type*, *@subtype*) att.damaged (*@extent*, *@hand*, *@agent*, *@degree*, *@group*)

Used by model.pPart.transcriptional

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address bibl biblStruct binaryObject cb choice cit corr date del desc distinct email emph expan foreign gap gloss graphic hi index label lb list listBibl measure measureGrp mentioned milestone name note num orig pb ptr q quote ref reg rs said sic soCalled stage term time title unclear

dictionaries: lang oRef oVar pRef pVar

drama: camera caption castList move sound tech view

figures: figure formula table

gaiji: g

header: biblFull

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material msDesc origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName listNym listOrg listPerson listPlace nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att classSpec code eg egXML elementSpec gi ident listRef macroSpec moduleRef moduleSpec specDesc specGrp specGrpRef specList tag val

textcrit: app listWit witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element damage
{
   att.global.attributes,
   att.typed.attributes,
   att.damaged.attributes,
   macro.paraContent}
```

Example

```
<l>The Moving Finger wri<damage  agent="water"  group="1">es; and</damage> having writ,</l>
<l>Moves <damage  agent="water"  group="1">
    <supplied>on: nor all your</supplied>
  </damage> Piety nor Wit</l>
```

Note Since damage to text witnesses frequently makes them harder to read, the <damage> element will often contain an <unclear> element. If the damaged area is not continuous (e.g. a stain affecting several strings of text), the **group** attribute may be used to group together several related <damage> elements; alternatively the <join> element may be used to indicate which <damage> and <unclear> elements are part of the same physical phenomenon.The <damage>, <gap>, <del>, <unclear> and <supplied> elements may be closely allied in use. See section *11.5.2. Use of the <gap>, <del>, <damage>, <unclear>, and <supplied> Elements in Combination* for discussion of which element is appropriate for which circumstance.

<damageSpan/>

(damaged span of text) marks the beginning of a longer sequence of text which is damaged in some way but still legible.

Module transcr — *11. Representation of Primary Sources*

Attributes att.damaged (*@extent*, *@hand*, *@agent*, *@degree*, *@group*) att.typed (*@type*, *@subtype*) att.spanning (*@spanTo*)

Used by model.global.edit

May contain Empty element

Declaration

```
element damageSpan
{
   att.global.attributes,
   att.damaged.attributes,
   att.typed.attributes,
```

```
   att.spanning.attributes,
   empty
}
```

<sch:pattern name="spanTo_required_for_damageSpan"> <sch:rule context="tei:damageSpan"> <sch:assert test="@spanTo">The spanTo= attribute of <sch:name/> is required.</sch:assert> </sch:rule> </sch:pattern>

Example

```
<p>Paragraph partially damaged. This is the undamaged
portion <damageSpan  spanTo="#a34"/>and this the damaged
portion of the paragraph.</p>
<p>This paragraph is entirely damaged.</p>
<p>Paragraph partially damaged; in the middle of this
paragraph the damage ends and the anchor point marks
the start of the <anchor  xml:id="a34"/> undamaged part of the text. ...</p>
```

Note Both the beginning and ending of the damaged sequence must be marked: the beginning by the <delSpan> element, the ending by the target of the **spanTo** attribute: if no other element available, the <anchor> element may be used for this purpose.The damaged text must be at least partially legible, in order for the encoder to be able to transcribe it. If it is not legible at all, the <damageSpan> element should not be used. Rather, the <gap> or <unclear> element should be employed, with the value of the **reason** attribute giving the cause. See further sections *11.5.1. Damage, Illegibility, and Supplied Text* and *11.5.2. Use of the <gap>, <del>, <damage>, <unclear>, and <supplied> Elements in Combination*.

<datatype>

specifies the declared value for an attribute, by referring to any datatype defined by the chosen schema language.

Module tagdocs — *22. Documentation Elements*

Attributes In addition to global attributes

@minOccurs (minimum number of occurences) indicates the minimum number of times this datatype may occur in the specification of the attribute being defined
Status Optional
Datatype `data.count`

@maxOccurs (maximum number of occurences) indicates the maximum number of times this datatype may occur in the specification of the attribute being defined
Status Optional
Datatype `data.count | "unbounded"`

Used by attDef

May contain Empty element

Declaration

```
element datatype
{
   att.global.attributes,
   attribute minOccurs { data.count }?,
   attribute maxOccurs { data.count | "unbounded" }?,
```

```
   macro.schemaPattern*
}
```

Example

```
<datatype>
  <rng:data  type="token"/>
</datatype>
```

Example

```
<datatype>
  <rng:ref  name="data.enumerated"/>
</datatype>
```

Example The encoding in the following example requires that the attribute being defined contain at least two URIs in its value, as is the case for the **targets** attribute of <join>.

```
<datatype  minOccurs="2"  maxOccurs="unbounded">
  <rng:ref  name="data.pointer"/>
</datatype>
```

Note In the TEI scheme, most datatypes are expressed using pre-defined TEI macros, which map a name in the form `data.xxxx` to a RelaxNG or WSD defined datatype.

<date> contains a date in any format.

Module core — *3. Elements Available in All TEI Documents*

Attributes att.datableatt.datable.w3c (*@period*, *@when*, *@notBefore*, *@notAfter*, *@from*, *@to*) att.datable.iso (*@when-iso*, *@notBefore-iso*, *@notAfter-iso*, *@from-iso*, *@to-iso*) att.durationatt.duration.w3c (*@dur*) att.duration.iso (*@dur-iso*) att.editLike (*@cert*, *@resp*, *@evidence*, *@source*, *@precision*) att.typed (*@type*, *@subtype*)

@calendar indicates the system or calendar to which the date belongs.

Status Optional

Datatype `data.enumerated`

Suggested values include: **Gregorian** Gregorian calendar

Julian Julian calendar

Islamic Islamic or Muslim (hijri) lunar calendar

Hebrew Hebrew or Jewish lunisolar calendar

Revolutionary French Revolutionary calendar

Iranian Iranian or Persian (Jalaali) solar calendar

Coptic Coptic or Alexandrian calendar

Chinese Chinese lunisolar calendar

```
He was born on
<date
   calendar="Gregorian">Feb. 22, 1732</date>
(<date
```

```
                    calendar="Julian"
                    when="1732-02-22"> Feb. 11, 1731/32, O.S.</date>).
```

Used by model.dateLike model.publicationStmtPart

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address binaryObject cb choice corr date del distinct email emph expan foreign gap gloss graphic hi index lb measure measureGrp mentioned milestone name note num orig pb ptr ref reg rs sic soCalled term time title unclear

dictionaries: lang oRef oVar pRef pVar

figures: formula

gaiji: g

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident specDesc specList tag val

textcrit: app witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element date
{
   att.global.attributes,
   att.datable.w3c.attributes,
   att.datable.iso.attributes,
   att.duration.w3c.attributes,
   att.duration.iso.attributes,
   att.editLike.attributes,
   att.typed.attributes,
   attribute calendar
   {
      "Gregorian"
    | "Julian"
    | "Islamic"
    | "Hebrew"
    | "Revolutionary"
    | "Iranian"
    | "Coptic"
    | "Chinese"
    | xsd:Name
   }?,
```

```
   ( text | model.gLike | model.phrase | model.global )*
}
```

Example

```
<date when="1980-02">early February 1980</date>
```

Example

```
Given on the <date when="1977-06-12">Twelfth Day of June
in the Year of Our Lord One Thousand Nine Hundred and
Seventy-seven of the Republic the Two Hundredth and first
and of the University the Eighty-Sixth.</date>
```

Example

```
<date when="1990-09">September 1990</date>
```

<dateline> contains a brief description of the place, date, time, etc. of production of a letter, newspaper story, or other work, prefixed or suffixed to it as a kind of heading or trailer.

Module textstructure — *4. Default Text Structure*

Attributes Global attributes only

Used by closer opener model.divWrapper

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address binaryObject cb choice corr date del distinct email emph expan foreign gap gloss graphic hi index lb measure measureGrp mentioned milestone name note num orig pb ptr ref reg rs sic soCalled term time title unclear

dictionaries: lang oRef oVar pRef pVar

figures: formula

gaiji: g

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident specDesc specList tag val

textcrit: app witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element dateline { att.global.attributes, macro.phraseSeq }
```

Example

```
<dateline>Walden, this 29. of August 1592</dateline>
```

Example

```
<div type="chapter">
  <p>and his heart was going like mad and yes
     I said yes I will Yes.</p>
  <closer>
    <dateline>
      <name type="place">Trieste-Zürich-Paris,</name>
      <date>1914–1921</date>
    </dateline>
  </closer>
</div>
```

<death> contains information about a person's death, such as its date and place.

Module namesdates — *13. Names, Dates, People, and Places*

Attributes att.editLike (*@cert*, *@resp*, *@evidence*, *@source*, *@precision*) att.datableatt.datable.w3c (*@period*, *@when*, *@notBefore*, *@notAfter*, *@from*, *@to*) att.datable.iso (*@when-iso*, *@notBefore-iso*, *@notAfter-iso*, *@from-iso*, *@to-iso*) att.naming (*@key*, *@ref*, *@nymRef*)

Used by model.persEventLike

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address binaryObject cb choice corr date del distinct email emph expan foreign gap gloss graphic hi index lb measure measureGrp mentioned milestone name note num orig pb ptr ref reg rs sic soCalled term time title unclear

dictionaries: lang oRef oVar pRef pVar

figures: formula

gaiji: g

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident specDesc specList tag val

textcrit: app witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element death
{
   att.global.attributes,
   att.editLike.attributes,
   att.datable.w3c.attributes,
   att.datable.iso.attributes,
   att.naming.attributes,
   macro.phraseSeq}
```

Example

```
<death when="1902-10-01"/>
```

Example

```
<death when="1960-12-10">Passed away near
<name type="place">Aix-la-Chapelle</name>,
after suffering from cerebral palsy.
</death>
```

<decoDesc> (decoration description) contains a description of the decoration of a manuscript, either as a sequence of paragraphs, or as a sequence of topically organised <decoNote> elements.

Module msdescription — *10. Manuscript Description*

Attributes Global attributes only

Used by model.physDescPart

May contain

core: p

linking: ab

msdescription: decoNote

Declaration

```
element decoDesc { att.global.attributes, ( model.pLike+ | decoNote+ ) }
```

Example

```
<decoDesc>
  <p>The start of each book of the Bible with a 10-line historiated
     illuminated initial; prefaces decorated with 6-line blue initials with red
     penwork flourishing; chapters marked by 3-line plain red initials; verses
     with 1-line initials, alternately blue or red.</p>
</decoDesc>
```

<decoNote> (note on decoration) contains a note describing either a decorative component of a manuscript, or a fairly homogenous class of such components.

Module msdescription — *10. Manuscript Description*

Attributes att.typed (*@type*, *@subtype*)

Used by binding bindingDesc decoDesc msItemStruct seal sealDesc model.msItemPart

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address bibl biblStruct binaryObject cb choice cit corr date del desc distinct email emph expan foreign gap gloss graphic hi index l label lb lg list listBibl measure measureGrp mentioned milestone name note num orig p pb ptr q quote ref reg rs said sic soCalled sp stage term time title unclear

dictionaries: lang oRef oVar pRef pVar

drama: camera caption castList move sound tech view

figures: figure formula table

gaiji: g

header: biblFull

iso-fs: fLib fs fvLib

linking: ab alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material msDesc origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName listNym listOrg listPerson listPlace nameLink offset orgName persName placeName region roleName settlement state surname

nets: eTree forest forestGrp graph tree

spoken: incident kinesic pause shift u vocal writing

tagdocs: att classSpec code eg egXML elementSpec gi ident listRef macroSpec moduleRef moduleSpec schemaSpec specDesc specGrp specGrpRef specList tag val

textcrit: app listWit witDetail

textstructure: floatingText

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element decoNote
{
   att.global.attributes,
   att.typed.attributes,
   macro.specialPara}
```

Example

```
<decoDesc>
  <decoNote  type="initial">
    <p>The start of each book of the Bible with
        a 10-line historiated illuminated initial;
        prefaces decorated with 6-line blue initials
        with red penwork flourishing; chapters marked by
```

```
            3-line plain red initials; verses with 1-line initials,
            alternately blue or red.</p>
        </decoNote>
    </decoDesc>
```

<def> (definition) contains definition text in a dictionary entry.

Module dictionaries — *9. Dictionaries*

Attributes att.lexicographic (*@expand*, *@norm*, *@split*, *@value*, *@orig*, *@location*, *@mergedIn*, *@opt*)

Used by etym model.entryPart.top model.entryPart

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address bibl biblStruct binaryObject cb choice cit corr date del desc distinct email emph expan foreign gap gloss graphic hi index label lb list listBibl measure measureGrp mentioned milestone name note num orig pb ptr q quote ref reg rs said sic soCalled stage term time title unclear

dictionaries: lang oRef oVar pRef pVar

drama: camera caption castList move sound tech view

figures: figure formula table

gaiji: g

header: biblFull

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material msDesc origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName listNym listOrg listPerson listPlace nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att classSpec code eg egXML elementSpec gi ident listRef macroSpec moduleRef moduleSpec specDesc specGrp specGrpRef specList tag val

textcrit: app listWit witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element def
{
   att.global.attributes,
   att.lexicographic.attributes,
   macro.paraContent}
```

Example

```
<entry>
  <form>
    <orth>competitor</orth>
    <hyph>com|peti|tor</hyph>
    <pron>k@m"petit@(r)</pron>
  </form>
  <gramGrp>
    <pos>n</pos>
  </gramGrp>
  <def>person who competes.</def>
</entry>
```

<default/> (default feature value) represents the value part of a feature-value specification which contains a defaulted value.

Module iso-fs — *18. Feature Structures*

Attributes Global attributes only

Used by model.featureVal.single

May contain Empty element

Declaration `element default { att.global.attributes, empty }`

Example

```
<f  name="gender">
  <default/>
</f>
```

<defaultVal> (default value) specifies the default declared value for an attribute.

Module tagdocs — *22. Documentation Elements*

Attributes Global attributes only

Used by attDef

May contain Character data only

Declaration `element defaultVal { att.global.attributes, text }`

Example

```
<defaultVal>#IMPLIED</defaultVal>
```

Note any legal declared value or TEI-defined keyword

<del> (deletion) contains a letter, word, or passage deleted, marked as deleted, or otherwise indicated as superfluous or spurious in the copy text by an author, scribe, annotator, or corrector.

Module core — *3. Elements Available in All TEI Documents*

Attributes att.transcriptional (*@hand*, *@status*, *@seq*) att.editLike (*@cert*, *@resp*, *@evidence*, *@source*, *@precision*) att.typed (*@type*, *@subtype*)

Used by model.pPart.transcriptional

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address bibl biblStruct binaryObject cb choice cit corr date del desc distinct email emph expan foreign gap gloss graphic hi index label lb list listBibl measure measureGrp mentioned milestone name note num orig pb ptr q quote ref reg rs said sic soCalled stage term time title unclear

dictionaries: lang oRef oVar pRef pVar

drama: camera caption castList move sound tech view

figures: figure formula table

gaiji: g

header: biblFull

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material msDesc origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName listNym listOrg listPerson listPlace nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att classSpec code eg egXML elementSpec gi ident listRef macroSpec moduleRef moduleSpec specDesc specGrp specGrpRef specList tag val

textcrit: app listWit witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element del
{
   att.global.attributes,
   att.transcriptional.attributes,
   att.editLike.attributes,
   att.typed.attributes,
   macro.paraContent}
```

Example

```
<l>
  <del  rend="overtyped">Mein</del> Frisch
<del  rend="overstrike"  type="primary">schwebt</del> weht der Wind
</l>
```

Note Degrees of uncertainty over what can still be read may be indicated by use of the <certainty> element (see *21. Certainty and Responsibility*).This element should be used for deletion of shorter sequences of text, typically single words or phrases. The <delSpan> element should be used for longer sequences of text, for those containing structural subdivisions, and for those containing overlapping additions and deletions.The text deleted must be at least partially legible, in order for

the encoder to be able to transcribe it. Illegible text within a deletion may be marked using the <gap> tag to signal that text is present but has not been transcribed. Attributes on the <gap> element may be used to indicate how much text is omitted, the reason for omitting it, etc. If text is not fully legible, the <unclear> element (available when using the additional tagset for transcription of primary sources) should be used to signal the areas of text which cannot be read with confidence in a similar way. See further sections *11.3.7. Text Omitted from or Supplied in the Transcription* and, for the close association of the <del> tag with the <gap>, <damage>, <unclear> and <supplied> elements (the latter three tags available when using the additional tagset for transcription of primary sources), *11.5.2. Use of the <gap>, <del>, <damage>, <unclear>, and <supplied> Elements in Combination.*The <del> tag should not be used for deletions made by editors or encoders. In these cases, either the <corr> tag or the <gap> tag should be used.

<delSpan/> (deleted span of text) marks the beginning of a longer sequence of text deleted, marked as deleted, or otherwise signaled as superfluous or spurious by an author, scribe, annotator, or corrector.

Module transcr — *11. Representation of Primary Sources*

Attributes att.transcriptional (*@hand*, *@status*, *@seq*) att.editLike (*@cert*, *@resp*, *@evidence*, *@source*, *@precision*) att.typed (*@type*, *@subtype*) att.spanning (*@spanTo*)

Used by model.global.edit

May contain Empty element

Declaration

```
element delSpan
{
   att.global.attributes,
   att.transcriptional.attributes,
   att.editLike.attributes,
   att.typed.attributes,
   att.spanning.attributes,
   empty
}
```

<sch:pattern name="spanTo_required_for_delSpan"> <sch:rule context="tei:delSpan"> <sch:assert test="@spanTo">The spanTo= attribute of <sch:name/> is required.</sch:assert> </sch:rule> </sch:pattern>

Example

```
<p>Paragraph partially deleted. This is the undeleted
portion <delSpan  spanTo="#a23"/>and this the deleted
portion of the paragraph.</p>
<p>Paragraph deleted together with adjacent material.</p>
<p>Second fully deleted paragraph.</p>
<p>Paragraph partially deleted; in the middle of this
paragraph the deletion ends and the anchor point marks
the resumption <anchor  xml:id="a23"/> of the text. ...</p>
```

Note Both the beginning and ending of the deleted sequence must be marked: the beginning by the <delSpan> element, the ending by the target of the **spanTo** attribute.The text deleted must be at least partially legible, in order for the encoder to be able to transcribe it. If it is not legible at all, the <delSpan> tag should not be used. Rather, the <gap> tag should be employed to signal that

text cannot be transcribed, with the value of the **reason** attribute giving the cause for the omission from the transcription as deletion. If it is not fully legible, the <unclear> element should be used to signal the areas of text which cannot be read with confidence. See further sections *11.3.7. Text Omitted from or Supplied in the Transcription* and, for the close association of the <delSpan> tag with the <gap>, <damage>, <unclear> and <supplied> elements, *11.5.2. Use of the <gap>, <del>, <damage>, <unclear>, and <supplied> Elements in Combination.*The <delSpan> tag should not be used for deletions made by editors or encoders. In these cases, either the <corr> tag or the <gap> tag should be used.

<depth> specifies a length measured across the spine.

Module msdescription — *10. Manuscript Description*

Attributes att.dimensions (*@unit*, *@quantity*, *@precision*, *@scope*)

Used by dimensions model.measureLike

May contain

gaiji: g

Declaration

```
element depth { att.global.attributes, att.dimensions.attributes, macro.xtext }
```

Example

```
<depth unit="in" quantity="4"/>
```

<derivation> describes the nature and extent of originality of this text.

Module corpus — *15. Language Corpora*

Attributes In addition to global attributes

@type categorizes the derivation of the text.

Status Optional

Datatype `data.enumerated`

Sample values include: **original** text is original

revision text is a revision of some other text

translation text is a translation of some other text

abridgment text is an abridged version of some other text

plagiarism text is plagiarized from some other text

traditional text has no obvious source but is one of a number derived from some common ancestor

Used by model.textDescPart

May contain

analysis: interp interpGrp span spanGrp

certainty: certainty respons

core: abbr address cb choice date distinct email emph expan foreign gap gloss index lb measure measureGrp mentioned milestone name note num pb ptr ref rs soCalled term time title

dictionaries: lang

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident tag val

textcrit: witDetail

transcr: addSpan am damageSpan delSpan ex fw handShift space subst

Declaration

```
element derivation
{
   att.global.attributes,
   attribute type { data.enumerated }?,
   macro.phraseSeq.limited}
```

Example

```
<derivation type="original"/>
```

Note For derivative texts, details of the ancestor may be included in the source description.

<desc> (description) contains a brief description of the intended usage, purpose, or application of its parent element.

Module core — *3. Elements Available in All TEI Documents*

Attributes att.translatable (*@version*)

Used by charDecl relation model.glossLike model.labelLike

May contain

core: abbr address bibl biblStruct choice cit date desc distinct email emph expan foreign gloss label list listBibl measure measureGrp mentioned name num ptr q quote ref rs said soCalled stage term time title

dictionaries: lang

drama: camera caption castList move sound tech view

figures: figure table

header: biblFull

msdescription: catchwords depth dimensions height heraldry locus material msDesc origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName listNym listOrg listPerson listPlace nameLink offset orgName persName placeName region roleName settlement state surname

tagdocs: att classSpec code eg egXML elementSpec gi ident listRef macroSpec moduleRef moduleSpec specGrp specGrpRef tag val

textcrit: listWit

transcr: am ex handShift subst

Declaration

```
element desc
{
   att.global.attributes,
   att.translatable.attributes,
   macro.limitedContent}
```

Example

```
<desc>contains a brief description of the purpose and application for
an element, attribute, attribute value, class, or entity.</desc>
```

Note TEI convention requires that this be expressed as a finite clause, begining with an active verb.

<dictScrap>

(dictionary scrap) encloses a part of a dictionary entry in which other phrase-level dictionary elements are freely combined.

Module dictionaries — *9. Dictionaries*

Attributes Global attributes only

Used by superEntry model.entryPart.top

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address bibl biblStruct binaryObject cb choice cit corr date del desc distinct email emph expan foreign gap gloss graphic hi index label lb list listBibl measure measureGrp mentioned milestone name note num orig pb ptr q quote ref reg rs said sic soCalled stage term time title unclear

dictionaries: case colloc def etym form gen gramGrp hom hyph iType lang lbl mood number oRef oVar orth pRef pVar per pos pron re sense stress subc superEntry syll tns usg xr

drama: camera caption castList move sound tech view

figures: figure formula table

gaiji: g

header: biblFull

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material msDesc origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName listNym listOrg listPerson listPlace nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att classSpec code eg egXML elementSpec gi ident listRef macroSpec moduleRef moduleSpec specDesc specGrp specGrpRef specList tag val

textcrit: app listWit witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element dictScrap
{
   att.global.attributes,
   (
      text
    | model.gLike    | model.entryPart    | model.phrase    | model.inter    | model.global   )*
}
```

Example

```
<entry>
  <dictScrap>
    <orth>biryani</orth> or <orth>biriani</orth>
    <pron>(%bIrI"A:nI)</pron>
    <def>any of a variety of Indian dishes ...</def>
    <etym>[from <lang>Urdu</lang>]</etym>
  </dictScrap>
</entry>
```

Note May contain any dictionary elements in any combination.This element is used to mark part of a dictionary entry in which lower level dictionary elements appear, but which does not itself form an identifiable structural unit.

<dimensions> contains a dimensional specification.

Module msdescription — *10. Manuscript Description*

Attributes att.dimensions (*@unit*, *@quantity*, *@precision*, *@scope*)

@type indicates which aspect of the object is being measured.

Status Optional

Datatype `data.enumerated`

Sample values include: **leaves** dimensions relate to one or more leaves (e.g. a single leaf, a gathering, or a separately bound part)

ruled dimensions relate to the area of a leaf which has been ruled in preparation for writing.

pricked dimensions relate to the area of a leaf which has been pricked out in preparation for ruling (used where this differs significantly from the ruled area, or where the ruling is not measurable).

written dimensions relate to the area of a leaf which has been written, with the height measured from the top of the minims on the top line of writing, to the bottom of the minims on the bottom line of writing.

miniatures dimensions relate to the miniatures within the manuscript

binding dimensions relate to the binding in which the codex or manuscript is contained

box dimensions relate to the box or other container in which the manuscript is stored.

Used by model.pPart.msdesc

May contain

msdescription: depth height width

Declaration

```
element dimensions
{
   att.global.attributes,
   att.dimensions.attributes,
   attribute type { data.enumerated }?,
   ( height?, width?, depth? )
}
```

Example

```
<dimensions type="leaves">
  <height scope="range">157-160</height>
  <width>105</width>
</dimensions>
<dimensions type="ruled">
  <height scope="most">90</height>
  <width scope="most">48</width>
</dimensions>
<dimensions unit="in">
  <height>12</height>
  <width>10</width>
</dimensions>
```

Example When simple numeric quantities are involved, they may be expressed on the **quantity** attribute of any or all of the child elements, as in the following example.

```
<dimensions type="leaves">
  <height scope="range">157-160</height>
  <width quantity="105"/>
</dimensions>
<dimensions type="ruled">
  <height unit="cm" scope="most" quantity="90"/>
  <width unit="cm" scope="most" quantity="48"/>
</dimensions>
<dimensions unit="in">
  <height quantity="12"/>
  <width quantity="10"/>
</dimensions>
```

Note Contains the length of one or more of a 1-, 2-, or 3-dimensional object's height, width, and depth.

<distinct> identifies any word or phrase which is regarded as linguistically distinct, for example as archaic, technical, dialectal, non-preferred, etc., or as forming part of a sublanguage.

Module core — *3. Elements Available in All TEI Documents*

Attributes In addition to global attributes

@type specifies the sublanguage or register to which the word or phrase is being assigned

Status Optional

Datatype `data.enumerated`

Values a semi-open user-defined list

@time specifies how the phrase is distinct diachronically

Status Optional

Datatype `data.code`

Values a semi-open user-defined list

@space specifies how the phrase is distinct diatopically

Status Optional

Datatype `data.code`

Values a semi-open user-defined list

@social specifies how the phrase is distinct diastatically

Status Optional

Datatype `data.code`

Values a semi-open user-defined list

Used by model.emphLike

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address binaryObject cb choice corr date del distinct email emph expan foreign gap gloss graphic hi index lb measure measureGrp mentioned milestone name note num orig pb ptr ref reg rs sic soCalled term time title unclear

dictionaries: lang oRef oVar pRef pVar

figures: formula

gaiji: g

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident specDesc specList tag val

textcrit: app witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element distinct
{
```

```
  att.global.attributes,
  attribute type { data.enumerated }?,
  attribute time { data.code }?,
  attribute space { data.code }?,
  attribute social { data.code }?,
  macro.phraseSeq}
```

Example

```
Next morning a boy in that dormitory confided to his
bosom friend, a <distinct  type="ps_slang">fag</distinct> of
Macrea's, that there was trouble in their midst which
King <distinct  type="archaic">would fain</distinct> keep
secret.
```

Source: [111]

<distributor> supplies the name of a person or other agency responsible for the distribution of a text.

Module header — *2. The TEI Header*

Attributes Global attributes only

Used by model.imprintPart model.publicationStmtPart

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address binaryObject cb choice corr date del distinct email emph expan foreign gap gloss graphic hi index lb measure measureGrp mentioned milestone name note num orig pb ptr ref reg rs sic soCalled term time title unclear

dictionaries: lang oRef oVar pRef pVar

figures: formula

gaiji: g

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident specDesc specList tag val

textcrit: app witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element distributor { att.global.attributes, macro.phraseSeq }
```

Example

```
<distributor>Oxford Text Archive</distributor>
<distributor>Redwood and Burn Ltd</distributor>
```

<district> contains the name of any kind of subdivision of a settlement, such as a parish, ward, or other administrative or geographic unit.

Module namesdates — *13. Names, Dates, People, and Places*

Attributes att.naming (*@key*, *@ref*, *@nymRef*) att.typed (*@type*, *@subtype*) att.datableatt.datable.w3c (*@period*, *@when*, *@notBefore*, *@notAfter*, *@from*, *@to*) att.datable.iso (*@when-iso*, *@notBefore-iso*, *@notAfter-iso*, *@from-iso*, *@to-iso*)

Used by model.placeNamePart

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address binaryObject cb choice corr date del distinct email emph expan foreign gap gloss graphic hi index lb measure measureGrp mentioned milestone name note num orig pb ptr ref reg rs sic soCalled term time title unclear

dictionaries: lang oRef oVar pRef pVar

figures: formula

gaiji: g

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident specDesc specList tag val

textcrit: app witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element district
{
   att.global.attributes,
   att.naming.attributes,
   att.typed.attributes,
   att.datable.w3c.attributes,
   att.datable.iso.attributes,
   macro.phraseSeq}
```

Example

```
<placeName>
  <district type="ward">Jericho</district>
  <settlement>Oxford</settlement>
</placeName>
```

Example

```
<placeName>
  <district type="area">South Side</district>
  <settlement>Chicago</settlement>
</placeName>
```

** ** (text division) contains a subdivision of the front, body, or back of a text.

Module textstructure — *4. Default Text Structure*

Attributes att.divLike (*@org*, *@sample*, *@part*) att.metrical (*@met*, *@real*, *@rhyme*) att.typed (*@type*, *@subtype*) att.declaring (*@decls*)

Used by model.divLike

May contain

analysis: interp interpGrp span spanGrp

certainty: certainty respons

core: bibl biblStruct cb cit desc divGen gap head index l label lb lg list listBibl meeting milestone note p pb q quote said sp stage

dictionaries: entry entryFree superEntry

drama: camera caption castList move sound tech view

figures: figure table

header: biblFull

iso-fs: fLib fs fvLib

linking: ab alt altGrp anchor join joinGrp link linkGrp timeline

msdescription: msDesc

namesdates: listNym listOrg listPerson listPlace

nets: eTree forest forestGrp graph tree

spoken: incident kinesic pause shift u vocal writing

tagdocs: classSpec eg egXML elementSpec listRef macroSpec moduleRef moduleSpec schemaSpec specGrp specGrpRef

textcrit: listWit witDetail

textstructure: argument byline closer dateline div docAuthor docDate epigraph floatingText opener postscript salute signed trailer

transcr: addSpan damageSpan delSpan fw space

Declaration

```
element div
{
   att.global.attributes,
   att.divLike.attributes,
   att.metrical.attributes,
```

```
   att.typed.attributes,
   att.declaring.attributes,
   (
      ( model.divTop | model.global )*,
      (
         (
            ( ( ( model.divLike | model.divGenLike ), model.global* )+ )
          | (
               ( ( model.common ), model.global* )+,
               ( ( model.divLike | model.divGenLike ), model.global* )*
            )
         ),
         ( ( model.divBottom ), model.global* )*
      )?
   )
}
```

Example

```
<body>
  <div  type="part">
    <head>Fallacies of Authority</head>
    <p>The subject of which is Authority in various shapes, and the
        object, to repress all exercise of the reasoning faculty.</p>
    <div  n="1"  type="chapter">
      <head>The Nature of Authority</head>
      <p>With reference to any proposed measures having for their
           object the greatest happiness of the greatest number....</p>
      <div  n="1.1"  type="section">
        <head>Analysis of Authority</head>
        <p>What on any given occasion is the legitimate weight or
              influence to be attached to authority ... </p>
      </div>
      <div  n="1.2"  type="section">
        <head>Appeal to Authority, in What Cases Fallacious.</head>
        <p>Reference to authority is open to the charge of fallacy
              when... </p>
      </div>
    </div>
  </div>
</body>
```

<div1> (level-1 text division) contains a first-level subdivision of the front, body, or back of a text.

Module textstructure — *4. Default Text Structure*

Attributes att.divLike (*@org*, *@sample*, *@part*) att.metrical (*@met*, *@real*, *@rhyme*) att.typed (*@type*, *@subtype*) att.declaring (*@decls*)

Used by model.div1Like

May contain

analysis: interp interpGrp span spanGrp

certainty: certainty respons

core: bibl biblStruct cb cit desc divGen gap head index l label lb lg list listBibl meeting milestone note p pb q quote said sp stage

dictionaries: entry entryFree superEntry

drama: camera caption castList move sound tech view

figures: figure table

header: biblFull

iso-fs: fLib fs fvLib

linking: ab alt altGrp anchor join joinGrp link linkGrp timeline

msdescription: msDesc

namesdates: listNym listOrg listPerson listPlace

nets: eTree forest forestGrp graph tree

spoken: incident kinesic pause shift u vocal writing

tagdocs: classSpec eg egXML elementSpec listRef macroSpec moduleRef moduleSpec schemaSpec specGrp specGrpRef

textcrit: listWit witDetail

textstructure: argument byline closer dateline div2 docAuthor docDate epigraph floatingText opener postscript salute signed trailer

transcr: addSpan damageSpan delSpan fw space

Declaration

```
element div1
{
   att.global.attributes,
   att.divLike.attributes,
   att.metrical.attributes,
   att.typed.attributes,
   att.declaring.attributes,
   (
      ( model.divTop | model.global )*,
      (
         (
            ( ( model.div2Like | model.divGenLike ), model.global* )+
          | (
               ( ( model.common ), model.global* )+,
               ( ( model.div2Like | model.divGenLike ), model.global* )*
            )
         ),
         ( ( model.divBottom ), model.global* )*
      )?
   )
}
```

Example

```
<div1 xml:id="levi" n="I" type="part">
  <head>Part I: Of Man </head>
  <div2 xml:id="levi1" n="1" type="chapter">
    <head>Chap. I. Of Sense </head>
    <p>Concerning the Thoughts of man... </p>
  </div2>
</div1>
<div1 xml:id="levii" n="II" type="part">
```

```
  <head>Part II: Of Common-Wealth</head>
</div1>
```

Note any sequence of low-level structural elements, possibly grouped into lower subdivisions.

<div2>

(level-2 text division) contains a second-level subdivision of the front, body, or back of a text.

Module textstructure — *4. Default Text Structure*

Attributes att.divLike (*@org*, *@sample*, *@part*) att.metrical (*@met*, *@real*, *@rhyme*) att.typed (*@type*, *@subtype*) att.declaring (*@decls*)

Used by model.div2Like

May contain

analysis: interp interpGrp span spanGrp

certainty: certainty respons

core: bibl biblStruct cb cit desc divGen gap head index l label lb lg list listBibl meeting milestone note p pb q quote said sp stage

dictionaries: entry entryFree superEntry

drama: camera caption castList move sound tech view

figures: figure table

header: biblFull

iso-fs: fLib fs fvLib

linking: ab alt altGrp anchor join joinGrp link linkGrp timeline

msdescription: msDesc

namesdates: listNym listOrg listPerson listPlace

nets: eTree forest forestGrp graph tree

spoken: incident kinesic pause shift u vocal writing

tagdocs: classSpec eg egXML elementSpec listRef macroSpec moduleRef moduleSpec schemaSpec specGrp specGrpRef

textcrit: listWit witDetail

textstructure: argument byline closer dateline div3 docAuthor docDate epigraph floatingText opener postscript salute signed trailer

transcr: addSpan damageSpan delSpan fw space

Declaration

```
element div2
{
   att.global.attributes,
   att.divLike.attributes,
   att.metrical.attributes,
   att.typed.attributes,
   att.declaring.attributes,
   (
      ( model.divTop | model.global )*,
      (
         (
            ( ( model.div3Like | model.divGenLike ), model.global* )+
          | (
```

```
                ( ( model.common ), model.global* )+,
                ( ( model.div3Like | model.divGenLike ), model.global* )*
             )
          ),
          ( ( model.divBottom ), model.global* )*
       )?
    )
}
```

Example

```
<div1 n="2" type="part">
  <head>The Second Partition:
     The Cure of Melancholy</head>
  <div2 n="2.1" type="section">
    <div3 n="2.1.1" type="member">
      <div4 n="2.1.1.1" type="subsection">
        <head>Unlawful Cures rejected.</head>
        <p>Inveterate melancholy, howsoever it may seem to
                be a continuate, inexorable disease, hard to be
                cured, accompanying them to their graves most part
                (as <ref target="#a">Montanus</ref> observes), yet many
                times it may be helped...
        </p>
      </div4>
    </div3>
  </div2>
  <div2 n="2.2" type="section">
    <div3 n="2.2.1" type="member">
      <head>Sect. II. Memb. I</head>
      <p/>
    </div3>
  </div2>
  <div2 n="2.3" type="section">
    <div3 n="2.3.1" type="member">
      <head>Sect. III. Memb. I</head>
      <p/>
    </div3>
  </div2>
</div1>
```

Note any sequence of low-level structural elements, possibly grouped into lower subdivisions.

<div3> (level-3 text division) contains a third-level subdivision of the front, body, or back of a text.

Module textstructure — *4. Default Text Structure*

Attributes att.divLike (*@org*, *@sample*, *@part*) att.metrical (*@met*, *@real*, *@rhyme*) att.typed (*@type*, *@subtype*) att.declaring (*@decls*)

Used by model.div3Like

May contain

analysis: interp interpGrp span spanGrp

certainty: certainty respons

core: bibl biblStruct cb cit desc divGen gap head index l label lb lg list listBibl meeting milestone note p pb q quote said sp stage

dictionaries: entry entryFree superEntry

drama: camera caption castList move sound tech view

figures: figure table

header: biblFull

iso-fs: fLib fs fvLib

linking: ab alt altGrp anchor join joinGrp link linkGrp timeline

msdescription: msDesc

namesdates: listNym listOrg listPerson listPlace

nets: eTree forest forestGrp graph tree

spoken: incident kinesic pause shift u vocal writing

tagdocs: classSpec eg egXML elementSpec listRef macroSpec moduleRef moduleSpec schemaSpec specGrp specGrpRef

textcrit: listWit witDetail

textstructure: argument byline closer dateline div4 docAuthor docDate epigraph floatingText opener postscript salute signed trailer

transcr: addSpan damageSpan delSpan fw space

Declaration

```
element div3
{
   att.global.attributes,
   att.divLike.attributes,
   att.metrical.attributes,
   att.typed.attributes,
   att.declaring.attributes,
   (
      ( model.divTop | model.global )*,
      (
         (
            ( ( model.div4Like | model.divGenLike ), model.global* )+
          | (
               ( ( model.common ), model.global* )+,
               ( ( model.div4Like | model.divGenLike ), model.global* )*
            )
         ),
         ( ( model.divBottom ), model.global* )*
      )?
   )
}
```

Example

```
<div2  n="2.2"  type="section">
  <div3  n="2.2.1"  type="member">
    <head>Sect. II. Memb. I</head>
    <p/>
  </div3>
  <div3  n="2.2.2"  type="member">
    <head>Memb. II
        Retention and Evacuation rectified.</head>
    <p/>
  </div3>
```

```
  <div3  n="2.2.3"  type="member">
    <head>Memb.  III
        Ayr rectified.  With a digression of the Ayr.</head>
    <p/>
  </div3>
</div2>
```

Note any sequence of low-level structural elements, possibly grouped into lower subdivisions.

<div4> (level-4 text division) contains a fourth-level subdivision of the front, body, or back of a text.

Module textstructure — *4. Default Text Structure*

Attributes att.divLike (*@org*, *@sample*, *@part*) att.metrical (*@met*, *@real*, *@rhyme*) att.typed (*@type*, *@subtype*) att.declaring (*@decls*)

Used by model.div4Like

May contain

analysis: interp interpGrp span spanGrp

certainty: certainty respons

core: bibl biblStruct cb cit desc divGen gap head index l label lb lg list listBibl meeting milestone note p pb q quote said sp stage

dictionaries: entry entryFree superEntry

drama: camera caption castList move sound tech view

figures: figure table

header: biblFull

iso-fs: fLib fs fvLib

linking: ab alt altGrp anchor join joinGrp link linkGrp timeline

msdescription: msDesc

namesdates: listNym listOrg listPerson listPlace

nets: eTree forest forestGrp graph tree

spoken: incident kinesic pause shift u vocal writing

tagdocs: classSpec eg egXML elementSpec listRef macroSpec moduleRef moduleSpec schemaSpec specGrp specGrpRef

textcrit: listWit witDetail

textstructure: argument byline closer dateline div5 docAuthor docDate epigraph floatingText opener postscript salute signed trailer

transcr: addSpan damageSpan delSpan fw space

Declaration

```
element div4
{
   att.global.attributes,
   att.divLike.attributes,
   att.metrical.attributes,
   att.typed.attributes,
   att.declaring.attributes,
   (
      ( model.divTop | model.global )*,
```

```
      (
         (
            ( ( model.div5Like | model.divGenLike ), model.global* )+
          | (
               ( ( model.common ), model.global* )+,
               ( ( model.div5Like | model.divGenLike ), model.global* )*
            )
         ),
         ( ( model.divBottom ), model.global* )*
      )?
   )
}
```

Example

```
<div3  n="2.2.1"  type="member">
  <head>Sect. II. Memb. I</head>
  <div4  n="2.2.1.1"  type="subsection">
    <head>Subsect I. — Dyet rectified in substance.</head>
    <p>Diet, <term  xml:lang="grc">diaitotiku</term>,
    <term  xml:lang="la">victus</term> or living
    </p>
  </div4>
  <div4  n="2.2.2.1"  type="subsection">
    <head>Subsect II. — Dyet rectified in quantity.</head>
    <p>Man alone, saith Cardan, eates and drinks without
        appetite, and useth all his pleasures without necessity
    </p>
  </div4>
</div3>
```

Note any sequence of low-level structural elements, possibly grouped into lower subdivisions.

<div5> (level-5 text division) contains a fifth-level subdivision of the front, body, or back of a text.

Module textstructure — *4. Default Text Structure*

Attributes att.divLike (*@org*, *@sample*, *@part*) att.metrical (*@met*, *@real*, *@rhyme*) att.typed (*@type*, *@subtype*) att.declaring (*@decls*)

Used by model.div5Like

May contain

analysis: interp interpGrp span spanGrp

certainty: certainty respons

core: bibl biblStruct cb cit desc divGen gap head index l label lb lg list listBibl meeting milestone note p pb q quote said sp stage

dictionaries: entry entryFree superEntry

drama: camera caption castList move sound tech view

figures: figure table

header: biblFull

iso-fs: fLib fs fvLib

linking: ab alt altGrp anchor join joinGrp link linkGrp timeline

msdescription: msDesc

namesdates: listNym listOrg listPerson listPlace

nets: eTree forest forestGrp graph tree

spoken: incident kinesic pause shift u vocal writing

tagdocs: classSpec eg egXML elementSpec listRef macroSpec moduleRef moduleSpec schemaSpec specGrp specGrpRef

textcrit: listWit witDetail

textstructure: argument byline closer dateline div6 docAuthor docDate epigraph floatingText opener postscript salute signed trailer

transcr: addSpan damageSpan delSpan fw space

Declaration

```
element div5
{
   att.global.attributes,
   att.divLike.attributes,
   att.metrical.attributes,
   att.typed.attributes,
   att.declaring.attributes,
   (
      ( model.divTop | model.global )*,
      (
         (
            ( ( model.div6Like | model.divGenLike ), model.global* )+
          | (
               ( ( model.common ), model.global* )+,
               ( ( model.div6Like | model.divGenLike ), model.global* )*
            )
         ),
         ( ( model.divBottom ), model.global* )*
      )?
   )
}
```

Note any sequence of low-level structural elements, possibly grouped into lower subdivisions.

<div6> (level-6 text division) contains a sixth-level subdivision of the front, body, or back of a text.

Module textstructure — *4. Default Text Structure*

Attributes att.divLike (*@org*, *@sample*, *@part*) att.metrical (*@met*, *@real*, *@rhyme*) att.typed (*@type*, *@subtype*) att.declaring (*@decls*)

Used by model.div6Like

May contain

analysis: interp interpGrp span spanGrp

certainty: certainty respons

core: bibl biblStruct cb cit desc divGen gap head index l label lb lg list listBibl meeting milestone note p pb q quote said sp stage

dictionaries: entry entryFree superEntry

drama: camera caption castList move sound tech view

figures: figure table

header: biblFull

iso-fs: fLib fs fvLib

linking: ab alt altGrp anchor join joinGrp link linkGrp timeline

msdescription: msDesc

namesdates: listNym listOrg listPerson listPlace

nets: eTree forest forestGrp graph tree

spoken: incident kinesic pause shift u vocal writing

tagdocs: classSpec eg egXML elementSpec listRef macroSpec moduleRef moduleSpec schemaSpec specGrp specGrpRef

textcrit: listWit witDetail

textstructure: argument byline closer dateline div7 docAuthor docDate epigraph floatingText opener postscript salute signed trailer

transcr: addSpan damageSpan delSpan fw space

Declaration

```
element div6
{
   att.global.attributes,
   att.divLike.attributes,
   att.metrical.attributes,
   att.typed.attributes,
   att.declaring.attributes,
   (
      ( model.divTop | model.global )*,
      (
         (
            ( ( model.div7Like | model.divGenLike ), model.global* )+
          | (
               ( ( model.common ), model.global* )+,
               ( ( model.div7Like | model.divGenLike ), model.global* )*
            )
         ),
         ( ( model.divBottom ), model.global* )*
      )?
   )
}
```

Note any sequence of low-level structural elements, possibly grouped into lower subdivisions.

<div7> (level-7 text division) contains the smallest possible subdivision of the front, body or back of a text, larger than a paragraph.

Module textstructure — *4. Default Text Structure*

Attributes att.divLike (*@org*, *@sample*, *@part*) att.metrical (*@met*, *@real*, *@rhyme*) att.typed (*@type*, *@subtype*) att.declaring (*@decls*)

Used by model.div7Like

May contain

analysis: interp interpGrp span spanGrp

certainty: certainty respons

core: bibl biblStruct cb cit desc gap head index l label lb lg list listBibl meeting milestone note p pb q quote said sp stage

dictionaries: entry entryFree superEntry

drama: camera caption castList move sound tech view

figures: figure table

header: biblFull

iso-fs: fLib fs fvLib

linking: ab alt altGrp anchor join joinGrp link linkGrp timeline

msdescription: msDesc

namesdates: listNym listOrg listPerson listPlace

nets: eTree forest forestGrp graph tree

spoken: incident kinesic pause shift u vocal writing

tagdocs: classSpec eg egXML elementSpec listRef macroSpec moduleRef moduleSpec schemaSpec specGrp specGrpRef

textcrit: listWit witDetail

textstructure: argument byline closer dateline docAuthor docDate epigraph floatingText opener postscript salute signed trailer

transcr: addSpan damageSpan delSpan fw space

Declaration

```
element div7
{
   att.global.attributes,
   att.divLike.attributes,
   att.metrical.attributes,
   att.typed.attributes,
   att.declaring.attributes,
   (
      ( model.divTop | model.global )*,
      (
         ( ( model.common ), model.global* )+,
         ( ( model.divBottom ), model.global* )*
      )?
   )
}
```

Note any sequence of low-level structural elements, e.g., paragraphs (<p>), lists (<list>), or examples (<eg> or <egXML>).

<divGen> (automatically generated text division) indicates the location at which a textual division generated automatically by a text-processing application is to appear.

Module core — *3. Elements Available in All TEI Documents*

Attributes In addition to global attributes

@type specifies what type of generated text division (e.g. index, table of contents, etc.) is to appear.

Status Optional

Datatype `data.enumerated`

Sample values include: **index** an index is to be generated and inserted at this point.

toc a table of contents

figlist a list of figures

tablist a list of tables

Note Valid values are application-dependent; those shown are of obvious utility in document production, but are by no means exhaustive.

Used by model.frontPart model.divGenLike

May contain

core: head

Declaration

```
element divGen
{
   att.global.attributes,
   attribute type { data.enumerated }?,
   model.headLike*
}
```

Example One use for this element is to allow document preparation software to generate an index and insert it in the appropriate place in the output. The example below assumes that the **indexName** attribute on <index> elements in the text has been used to specify index entries for the two generated indexes, named NAMES and THINGS:

```
<back>
  <div1 type="backmat">
    <head>Bibliography</head>
    <listBibl>
      <bibl/>
    </listBibl>
  </div1>
  <div1 type="backmat">
    <head>Indices</head>
    <divGen n="Index Nominum" type="NAMES"/>
    <divGen n="Index Rerum" type="THINGS"/>
  </div1>
</back>
```

Example Another use for <divGen> is to specify the location of an automatically produced table of contents:

```
<front>
<!--<titlePage>...</titlePage>-->
  <divGen type="toc"/>
  <div>
    <head>Preface</head>
    <p> ... </p>
  </div>
</front>
```

Note This element is intended primarily for use in document production or manipulation, rather than in the transcription of pre-existing materials; it makes it easier to specify the location of indices, tables of contents, etc., to be generated by text preparation or word processing software.

<docAuthor> (document author) contains the name of the author of the document, as given on the title page (often but not always contained in a byline).

Module textstructure — *4. Default Text Structure*

Attributes Global attributes only

Used by byline model.titlepagePart model.divWrapper model.pLike.front

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address binaryObject cb choice corr date del distinct email emph expan foreign gap gloss graphic hi index lb measure measureGrp mentioned milestone name note num orig pb ptr ref reg rs sic soCalled term time title unclear

dictionaries: lang oRef oVar pRef pVar

figures: formula

gaiji: g

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident specDesc specList tag val

textcrit: app witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element docAuthor { att.global.attributes, macro.phraseSeq }
```

Example

```
<titlePage>
  <docTitle>
    <titlePart>Travels into Several Remote Nations of the World, in Four
        Parts.</titlePart>
  </docTitle>
  <byline> By <docAuthor>Lemuel Gulliver</docAuthor>, First a Surgeon,
     and then a Captain of several Ships</byline>
</titlePage>
```

Note The document author's name often occurs within a byline, but the <docAuthor> element may be used whether the <byline> element is used or not.

<docDate> (document date) contains the date of a document, as given (usually) on a title page.

Module textstructure — *4. Default Text Structure*

Attributes In addition to global attributes

@when gives the value of the date in standard form, i.e. YYYY-MM-DD.

Status Optional

Datatype `data.temporal.w3c`

Values a date in one of the formats specified in *XML Schema Part 2: Datatypes Second Edition*

Note For simple dates, the **when** attribute should give the Gregorian or proleptic Gregorian date in the form (YYYY-MM-DD) specified by *XML Schema Part 2*.

Used by docImprint model.titlepagePart model.divWrapper model.pLike.front

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address binaryObject cb choice corr date del distinct email emph expan foreign gap gloss graphic hi index lb measure measureGrp mentioned milestone name note num orig pb ptr ref reg rs sic soCalled term time title unclear

dictionaries: lang oRef oVar pRef pVar

figures: formula

gaiji: g

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident specDesc specList tag val

textcrit: app witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element docDate
{
   att.global.attributes,
   attribute when { data.temporal.w3c }?,
   macro.phraseSeq}
```

Example

```
<docImprint>Oxford, Clarendon Press, <docDate>1987</docDate>
</docImprint>
```

Note Cf. the general <date> element in the core tag set. This specialized element is provided for convenience in marking and processing the date of the documents, since it is likely to require specialized handling for many applications.

<docEdition> (document edition) contains an edition statement as presented on a title page of a document.

Module textstructure — *4. Default Text Structure*

Attributes Global attributes only

Used by model.titlepagePart model.pLike.front

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address bibl biblStruct binaryObject cb choice cit corr date del desc distinct email emph expan foreign gap gloss graphic hi index label lb list listBibl measure measureGrp mentioned milestone name note num orig pb ptr q quote ref reg rs said sic soCalled stage term time title unclear

dictionaries: lang oRef oVar pRef pVar

drama: camera caption castList move sound tech view

figures: figure formula table

gaiji: g

header: biblFull

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material msDesc origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName listNym listOrg listPerson listPlace nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att classSpec code eg egXML elementSpec gi ident listRef macroSpec moduleRef moduleSpec specDesc specGrp specGrpRef specList tag val

textcrit: app listWit witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element docEdition { att.global.attributes, macro.paraContent }
```

Example

```
<docEdition>The Third edition Corrected</docEdition>
```

Note Cf. the <edition> element of bibliographic citation. As usual, the shorter name has been given to the more frequent element.

<docImprint> (document imprint) contains the imprint statement (place and date of publication, publisher name), as given (usually) at the foot of a title page.

Module textstructure — *4. Default Text Structure*

Attributes Global attributes only

Used by model.titlepagePart model.pLike.front

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address binaryObject cb choice corr date del distinct email emph expan foreign gap gloss graphic hi index lb measure measureGrp mentioned milestone name note num orig pb ptr pubPlace publisher ref reg rs sic soCalled term time title unclear

dictionaries: lang oRef oVar pRef pVar

figures: formula

gaiji: g

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident specDesc specList tag val

textcrit: app witDetail

textstructure: docDate

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element docImprint
{
   att.global.attributes,
   (
      text
    | model.gLike    | model.phrase    | pubPlace    | docDate    | publisher    | model.global   )*
}
```

Example

```
<docImprint>Oxford, Clarendon Press, 1987</docImprint>
```

Imprints may be somewhat more complex:

```
<docImprint>
  <pubPlace>London</pubPlace>
Printed for <name>E. Nutt</name>,
at
<pubPlace>Royal Exchange</pubPlace>;
<name>J. Roberts</name> in
<pubPlace>wick-Lane</pubPlace>;
<name>A. Dodd</name> without
<pubPlace>Temple-Bar</pubPlace>;
and <name>J. Graves</name> in
<pubPlace>St. James's-street.</pubPlace>
  <date>1722.</date>
</docImprint>
```

Note Cf. the <imprint> element of bibliographic citations. As with title, author, and editions, the shorter name is reserved for the element likely to be used more often.

<docTitle> (document title) contains the title of a document, including all its constituents, as given on a title page.

Module textstructure — *4. Default Text Structure*

Attributes Global attributes only

Used by model.titlepagePart model.pLike.front

May contain

analysis: interp interpGrp span spanGrp

certainty: certainty respons

core: cb gap index lb milestone note pb

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp timeline

spoken: incident kinesic pause shift vocal writing

textcrit: witDetail

textstructure: titlePart

transcr: addSpan damageSpan delSpan fw space

Declaration

```
element docTitle
{
   att.global.attributes,
   ( model.global*, ( titlePart, model.global* )+ )
}
```

Example

```
<docTitle>
  <titlePart  type="main">The DUNCIAD,
     VARIOURVM.
  </titlePart>
  <titlePart  type="sub">WITH THE
     PROLEGOMENA of SCRIBLERUS.
```

```
      </titlePart>
    </docTitle>
```

<domain>

(domain of use) describes the most important social context in which the text was realized or for which it is intended, for example private vs. public, education, religion, etc.

Module corpus — *15. Language Corpora*

Attributes In addition to global attributes

@type categorizes the domain of use.

Status Optional

Datatype `data.enumerated`

Sample values include: **art** art and entertainment

domestic domestic and private

religious religious and ceremonial

business business and work place

education education

govt (government) government and law

public other forms of public context

Used by model.textDescPart

May contain

analysis: interp interpGrp span spanGrp

certainty: certainty respons

core: abbr address cb choice date distinct email emph expan foreign gap gloss index lb measure measureGrp mentioned milestone name note num pb ptr ref rs soCalled term time title

dictionaries: lang

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident tag val

textcrit: witDetail

transcr: addSpan am damageSpan delSpan ex fw handShift space subst

Declaration

```
element domain
{
   att.global.attributes,
```

```
   attribute type { data.enumerated }?,
   macro.phraseSeq.limited}
```

Example

```
<domain type="domestic"/>
<domain type="rel">religious broadcast</domain>
```

Note Usually empty, unless some further clarification of the type attribute is needed, in which case it may contain running prose.The list presented here is primarily for illustrative purposes.

<eLeaf> (leaf or terminal node of an embedding tree) provides explicitly for a leaf of an embedding tree, which may also be encoded with the eTree element.

Module nets — *19. Graphs, Networks, and Trees*

Attributes att.typed (*@type*, *@subtype*)

@value provides the value of an embedding leaf, which is a feature structure or other analytic element.

Status Required when applicable

Datatype `data.pointer`

Values A valid identifier of a feature structure or other analytic element.

Used by eTree triangle

May contain

core: label ptr ref

Declaration

```
element eLeaf
{
   att.global.attributes,
   att.typed.attributes,
   attribute value { data.pointer }?,
   ( label?, model.ptrLike? )
}
```

Example

```
<eLeaf value="http://an.fsurl.tei/#FSWITH">
  <label>with</label>
</eLeaf>
```

Note The <eTree> tag may be used if the encoder does not wish to distinguish by name between nonleaf and leaf nodes in embedding trees; they are distinguished by their arrangement.

<eTree> (embedding tree) provides an alternative to tree element for representing ordered rooted tree structures.

Module nets — *19. Graphs, Networks, and Trees*

Attributes att.typed (*@type*, *@subtype*)

@value provides the value of an embedding tree, which is a feature structure or other analytic element.

Status Required when applicable

Datatype `data.pointer`

Values A valid identifier of a feature structure or other analytic element.

Used by eTree forest triangle model.divPart

May contain

core: label ptr ref

nets: eLeaf eTree triangle

Declaration

```
element eTree
{
   att.global.attributes,
   att.typed.attributes,
   attribute value { data.pointer }?,
   ( label?, ( eTree | triangle | eLeaf | model.ptrLike )* )
}
```

Example

```
<eTree n="ex1">
  <label>PP</label>
  <eTree>
    <label>P</label>
    <eLeaf>
      <label>with</label>
    </eLeaf>
  </eTree>
  <eTree>
    <label>NP</label>
    <eTree>
      <label>Art</label>
      <eLeaf>
        <label>the</label>
      </eLeaf>
    </eTree>
    <eTree>
      <label>N</label>
      <eLeaf>
        <label>periscope</label>
      </eLeaf>
    </eTree>
  </eTree>
</eTree>
```

Note an optional label followed by zero or more embedding trees, triangles, or embedding leafs.

<edition>

(edition) describes the particularities of one edition of a text.

Module header — *2. The TEI Header*

Attributes Global attributes only

Used by editionStmt monogr model.biblPart

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address binaryObject cb choice corr date del distinct email emph expan foreign gap gloss graphic hi index lb measure measureGrp mentioned milestone name note num orig pb ptr ref reg rs sic soCalled term time title unclear

dictionaries: lang oRef oVar pRef pVar

figures: formula

gaiji: g

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident specDesc specList tag val

textcrit: app witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element edition { att.global.attributes, macro.phraseSeq }
```

Example

```
<edition>First edition <date>Oct 1990</date>
</edition>
<edition  n="S2">Students' edition</edition>
```

<editionStmt> (edition statement) groups information relating to one edition of a text.

Module header — *2. The TEI Header*

Attributes Global attributes only

Used by biblFull fileDesc

May contain

core: p respStmt

header: edition

linking: ab

Declaration

```
element editionStmt
{
   att.global.attributes,
```

```
    ( model.pLike+ | ( edition, respStmt* ) )
}
```

Example

```
<editionStmt>
  <edition  n="S2">Students' edition</edition>
  <respStmt>
    <resp>Adapted by </resp>
    <name>Elizabeth Kirk</name>
  </respStmt>
</editionStmt>
```

Example

```
<editionStmt>
  <p>First edition, <date>Michaelmas Term, 1991.</date>
  </p>
</editionStmt>
```

<editor> secondary statement of responsibility for a bibliographic item, for example the name of an individual, institution or organization, (or of several such) acting as editor, compiler, translator, etc.

Module core — *3. Elements Available in All TEI Documents*

Attributes In addition to global attributes

@role specifies the nature of the intellectual responsibility

Status Optional

Datatype `data.enumerated`

Values semi-open list (examples might include: translator, editor, compiler, illustrator, etc.)

Used by analytic monogr series titleStmt model.respLike

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address binaryObject cb choice corr date del distinct email emph expan foreign gap gloss graphic hi index lb measure measureGrp mentioned milestone name note num orig pb ptr ref reg rs sic soCalled term time title unclear

dictionaries: lang oRef oVar pRef pVar

figures: formula

gaiji: g

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident specDesc specList tag val

textcrit: app witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element editor
{
   att.global.attributes,
   attribute role { data.enumerated }?,
   macro.phraseSeq}
```

Example

```
<editor>Eric Johnson</editor>
<editor  role="illustrator">John Tenniel</editor>
```

Note A consistent format should be adoptedParticularly where cataloguing is likely to be based on the content of the header, it is advisable to use generally recognized authority lists for the exact form of personal names.

<editorialDecl>

<editorialDecl> (editorial practice declaration) provides details of editorial principles and practices applied during the encoding of a text.

Module header — *2. The TEI Header*

Attributes att.declarable (*@default*)

Used by model.encodingPart

May contain

core: p

header: correction hyphenation interpretation normalization quotation segmentation stdVals

linking: ab

Declaration

```
element editorialDecl
{
   att.global.attributes,
   att.declarable.attributes,
   ( model.pLike+ | model.editorialDeclPart+ )
}
```

Example

```
<editorialDecl>
  <normalization>
    <p>All words converted to Modern American spelling using
        Websters 9th Collegiate dictionary
    </p>
```

```
  </normalization>
  <quotation  marks="all"  form="std">
    <p>All opening quotation marks converted to “ all closing
        quotation marks converted to &cdq;.</p>
  </quotation>
</editorialDecl>
```

<education> contains a description of the educational experience of a person.

Module namesdates — *13. Names, Dates, People, and Places*

Attributes att.editLike (*@cert*, *@resp*, *@evidence*, *@source*, *@precision*) att.datableatt.datable.w3c (*@period*, *@when*, *@notBefore*, *@notAfter*, *@from*, *@to*) att.datable.iso (*@when-iso*, *@notBefore-iso*, *@notAfter-iso*, *@from-iso*, *@to-iso*) att.naming (*@key*, *@ref*, *@nymRef*)

Used by model.persStateLike

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address binaryObject cb choice corr date del distinct email emph expan foreign gap gloss graphic hi index lb measure measureGrp mentioned milestone name note num orig pb ptr ref reg rs sic soCalled term time title unclear

dictionaries: lang oRef oVar pRef pVar

figures: formula

gaiji: g

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident specDesc specList tag val

textcrit: app witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element education
{
   att.global.attributes,
   att.editLike.attributes,
   att.datable.w3c.attributes,
   att.datable.iso.attributes,
   att.naming.attributes,
   macro.phraseSeq}
```

Example

```
<education>Left school at age 16</education>
<education notBefore="1986-01-01" notAfter="1990-06-30">Attended
<name>Cherwell School</name>
</education>
```

<eg> (example) contains any kind of illustrative example.

Module tagdocs — *22. Documentation Elements*

Attributes att.xmlspace (*@xml:space*)

Used by exemplum model.egLike

May contain Character data only

Declaration

```
element eg { att.global.attributes, att.xmlspace.attributes, text }
```

Example

```
<p>The
<gi>term</gi> element is declared using the following syntax:
<eg><![CDATA[<!ELEMENT term (%phrase.content;)>]]</eg>
</p>
```

Note If the example contains material in XML markup, either it must be enclosed within a CDATA marked section, or character entity references must be used to represent the markup delimiters. If the example contains well-formed XML, it should be marked using the more specific <egXML> element.

<egXML> (example of XML) contains a single well-formed XML fragment demonstrating the use of some XML element or attribute, in which the <egXML> element itself functions as the root element.

Module tagdocs — *22. Documentation Elements*

Attributes att.xmlspace (*@xml:space*)

Used by exemplum model.egLike

May contain Character data only

Declaration

```
element egXML { att.global.attributes, att.xmlspace.attributes, text }
```

Example

```
<egXML><langUsage>
   <language ident="en">English</language>
</langUsage>
</egXML>
```

Note In the source of the TEI Guidelines, this element declares itself and its content as belonging to the namespace http://www.tei-c.org/ns/Examples. This enables the content of the element to be

validated independently against the TEI scheme. Where this element is used outside this context, a different namespace or none at all may be preferable. The content must however be a well-formed XML fragment or document: where this is not the case, the more general <eg> element should be used in preference.

<elementSpec> (element specification) documents the structure, content, and purpose of a single element type.

Module tagdocs — *22. Documentation Elements*

Attributes att.identified (*@ident*, *@predeclare*, *@module*, *@mode*)

@ns (namespace) specifies the namespace to which this element belongs

Status Optional

Datatype `data.namespace`

@usage specifies the optionality of an attribute or element.

Status Optional

Legal values are: **req** (required)

mwa (mandatory when applicable)

rec (recommended)

rwa (recommended when applicable)

opt (optional) **[Default]**

Used by model.oddDecl

May contain

core: desc gloss

tagdocs: altIdent attList classes content equiv exemplum listRef remarks

Declaration

```
element elementSpec
{
   att.global.attributes,
   att.identified.attributes,
   attribute ns { data.namespace }?,
   attribute usage { "req" | "mwa" | "rec" | "rwa" | "opt" }?,
   (
      model.glossLike*,
      classes?,
      content?,
      attList?,
      exemplum*,
      remarks*,
      listRef*
   )
}
```

Example

```
<elementSpec
   module="tagdocs"
   xml:id="Code"
   usage="mwa"
```

```
    ident="code">
  <equiv/>
  <gloss/>
  <desc>contains literal code</desc>
  <classes>
    <memberOf  key="model.emphLike"/>
  </classes>
  <content>
    <rng:text/>
  </content>
  <attList>
    <attDef  ident="type"  usage="opt">
      <equiv/>
      <desc>the language of the code</desc>
      <datatype>
        <rng:ref  name="data.enumerated"/>
      </datatype>
    </attDef>
  </attList>
</elementSpec>
```

<email> (electronic mail address) contains an e-mail address identifying a location to which e-mail messages can be delivered.

Module core — *3. Elements Available in All TEI Documents*

Attributes Global attributes only

Used by model.addressLike

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address binaryObject cb choice corr date del distinct email emph expan foreign gap gloss graphic hi index lb measure measureGrp mentioned milestone name note num orig pb ptr ref reg rs sic soCalled term time title unclear

dictionaries: lang oRef oVar pRef pVar

figures: formula

gaiji: g

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident specDesc specList tag val

textcrit: app witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element email { att.global.attributes, macro.phraseSeq }
```

Example

```
<email>editors@tei-c.org</email>
```

Note The format of a modern Internet email address is defined in RFC 2822

<emph> (emphasized) marks words or phrases which are stressed or emphasized for linguistic or rhetorical effect.

Module core — *3. Elements Available in All TEI Documents*

Attributes Global attributes only

Used by model.emphLike

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address bibl biblStruct binaryObject cb choice cit corr date del desc distinct email emph expan foreign gap gloss graphic hi index label lb list listBibl measure measureGrp mentioned milestone name note num orig pb ptr q quote ref reg rs said sic soCalled stage term time title unclear

dictionaries: lang oRef oVar pRef pVar

drama: camera caption castList move sound tech view

figures: figure formula table

gaiji: g

header: biblFull

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material msDesc origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName listNym listOrg listPerson listPlace nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att classSpec code eg egXML elementSpec gi ident listRef macroSpec moduleRef moduleSpec specDesc specGrp specGrpRef specList tag val

textcrit: app listWit witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element emph { att.global.attributes, macro.paraContent }
```

Example

```
You took the car and did <emph>what</emph>?!!
```

Example

```
<q>What it all comes to is this,</q> he said.
<q>
  <emph>What
     does Christopher Robin do in the morning nowadays?</emph>
</q>
```

Source: [142]

<encodingDesc> (encoding description) documents the relationship between an electronic text and the source or sources from which it was derived.

Module header — *2. The TEI Header*

Attributes Global attributes only

Used by model.headerPart

May contain

core: p

gaiji: charDecl

header: appInfo classDecl editorialDecl geoDecl projectDesc refsDecl samplingDecl tagsDecl

iso-fs: fsdDecl

linking: ab

textcrit: variantEncoding

verse: metDecl

Declaration

```
element encodingDesc
{
   att.global.attributes,
   ( ( model.encodingPart | model.pLike )+ )
}
```

Example

```
<encodingDesc>
  <p>Basic encoding, capturing lexical information only. All
     hyphenation, punctuation, and variant spellings normalized. No
     formatting or layout information preserved.</p>
</encodingDesc>
```

<entry> contains a reasonably well-structured dictionary entry.

Module dictionaries — *9. Dictionaries*

Attributes att.entryLike (*@type*, *@sortKey*)

Used by superEntry model.entryLike

May contain

analysis: interp interpGrp span spanGrp

certainty: certainty respons

core: cb cit gap index lb milestone note pb

dictionaries: def dictScrap etym form gramGrp hom re sense usg xr

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp timeline

spoken: incident kinesic pause shift vocal writing

textcrit: witDetail

transcr: addSpan damageSpan delSpan fw space

Declaration

```
element entry
{
   att.global.attributes,
   att.entryLike.attributes,
   ( hom | sense | model.entryPart.top | model.global )+
}
```

Example

```
<entry>
  <form>
    <orth>disproof</orth>
    <pron>dIs"pru:f</pron>
  </form>
  <gramGrp>
    <pos>n</pos>
  </gramGrp>
  <sense  n="1">
    <def>facts that disprove something.</def>
  </sense>
  <sense  n="2">
    <def>the act of disproving.</def>
  </sense>
</entry>
```

Note Like all elements, <entry> inherits an **xml:id** attribute from the class global. No restrictions are placed on the method used to construct **xml:id**s; one convenient method is to use the orthographic form of the headword, appending a disambiguating number where necessary. Identification codes are sometimes included on machine-readable tapes of dictionaries for in-house use.

<entryFree> (unstructured entry) contains a dictionary entry which does not necessarily conform to the constraints imposed by the <entry> element.

Module dictionaries — *9. Dictionaries*

Attributes att.entryLike (*@type*, *@sortKey*) att.lexicographic (*@expand*, *@norm*, *@split*, *@value*, *@orig*, *@location*, *@mergedIn*, *@opt*)

Used by model.entryLike

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address bibl biblStruct binaryObject cb choice cit corr date del desc distinct email emph expan foreign gap gloss graphic hi index label lb list listBibl measure measureGrp mentioned milestone name note num orig pb ptr q quote ref reg rs said sic soCalled stage term time title unclear

dictionaries: case colloc def etym form gen gramGrp hom hyph iType lang lbl mood number oRef oVar orth pRef pVar per pos pron re sense stress subc superEntry syll tns usg xr

drama: camera caption castList move sound tech view

figures: figure formula table

gaiji: g

header: biblFull

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material msDesc origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName listNym listOrg listPerson listPlace nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att classSpec code eg egXML elementSpec gi ident listRef macroSpec moduleRef moduleSpec specDesc specGrp specGrpRef specList tag val

textcrit: app listWit witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element entryFree
{
   att.global.attributes,
   att.entryLike.attributes,
   att.lexicographic.attributes,
   (
      text
    | model.gLike    | model.entryPart    | model.phrase    | model.inter    | model.global   )*
}
```

Example

```
<entryFree>
  <orth>biryani</orth> or <orth>biriani</orth>
  <pron>(%bIrI"A:nI)</pron>
```

```
  <def>any of a variety of Indian dishes ...</def>
  <etym>[from <lang>Urdu</lang>]</etym>
</entryFree>
```

Note May contain any dictionary elements in any combination.

<epigraph> contains a quotation, anonymous or attributed, appearing at the start of a section or chapter, or on a title page.

Module textstructure — *4. Default Text Structure*

Attributes Global attributes only

Used by opener model.divWrapper model.titlepagePart model.pLike.front

May contain

analysis: interp interpGrp span spanGrp

certainty: certainty respons

core: bibl biblStruct cb cit desc gap index l label lb lg list listBibl milestone note p pb q quote said sp stage

dictionaries: entry entryFree superEntry

drama: camera caption castList move sound tech view

figures: figure table

header: biblFull

iso-fs: fLib fs fvLib

linking: ab alt altGrp anchor join joinGrp link linkGrp timeline

msdescription: msDesc

namesdates: listNym listOrg listPerson listPlace

nets: eTree forest forestGrp graph tree

spoken: incident kinesic pause shift u vocal writing

tagdocs: classSpec eg egXML elementSpec listRef macroSpec moduleRef moduleSpec schemaSpec specGrp specGrpRef

textcrit: listWit witDetail

textstructure: floatingText

transcr: addSpan damageSpan delSpan fw space

Declaration

```
element epigraph { att.global.attributes, ( model.common | model.global )* }
```

Example

```
<epigraph  xml:lang="la">
  <cit>
    <bibl>Lucret.</bibl>
    <quote>
      <l  part="F">petere inde coronam,</l>
      <l>Vnde prius nulli velarint tempora Musae.</l>
    </quote>
```

```
    </cit>
  </epigraph>
```

<epilogue> contains the epilogue to a drama, typically spoken by an actor out of character, possibly in association with a particular performance or venue.

Module drama — *7. Performance Texts*

Attributes Global attributes only

Used by model.frontPart.drama

May contain

analysis: interp interpGrp span spanGrp

certainty: certainty respons

core: bibl biblStruct cb cit desc gap head index l label lb lg list listBibl meeting milestone note p pb q quote said sp stage

dictionaries: entry entryFree superEntry

drama: camera caption castList move sound tech view

figures: figure table

header: biblFull

iso-fs: fLib fs fvLib

linking: ab alt altGrp anchor join joinGrp link linkGrp timeline

msdescription: msDesc

namesdates: listNym listOrg listPerson listPlace

nets: eTree forest forestGrp graph tree

spoken: incident kinesic pause shift u vocal writing

tagdocs: classSpec eg egXML elementSpec listRef macroSpec moduleRef moduleSpec schemaSpec specGrp specGrpRef

textcrit: listWit witDetail

textstructure: argument byline closer dateline docAuthor docDate epigraph floatingText opener postscript salute signed trailer

transcr: addSpan damageSpan delSpan fw space

Declaration

```
element epilogue
{
   att.global.attributes,
   (
      ( model.divTop | model.global )*,
      ( ( model.common ), model.global* )+,
      ( ( model.divBottom ), model.global* )*
   )
}
```

Example

```
<epilogue>
  <head>Written by <name>Colley Cibber, Esq</name> and
     spoken by <name>Mrs. Cibber</name>
  </head>
  <sp>
    <lg  type="couplet">
      <l>Since Fate has robb'd me of the hapless Youth,</l>
      <l>For whom my heart had hoarded up its truth;</l>
    </lg>
    <lg  type="couplet">
      <l>By all the Laws of Love and Honour, now,</l>
      <l>I'm free again to chuse, — and one of you</l>
    </lg>
    <lg  type="triplet">
      <l>Suppose I search the sober Gallery; — No,</l>
      <l>There's none but Prentices — & Cuckolds all a row:</l>
      <l>And these, I doubt, are those that make 'em so.</l>
    </lg>
    <stage  type="business">Pointing to the Boxes.</stage>
    <lg  type="couplet">
      <l>'Tis very well, enjoy the jest:</l>
    </lg>
  </sp>
</epilogue>
```

Note Contains optional headings, a sequence of one or more component-level elements, and an optional sequence of closing material.

<equipment> provides technical details of the equipment and media used for an audio or video recording used as the source for a spoken text.

Module spoken — *8. Transcriptions of Speech*

Attributes att.declarable (*@default*)

Used by model.recordingPart

May contain

core: p

linking: ab

Declaration

```
element equipment
{
   att.global.attributes,
   att.declarable.attributes,
   model.pLike+
}
```

Example

```
<equipment>
  <p>"Hi-8" 8 mm NTSC camcorder with integral directional
     microphone and windshield and stereo digital sound
     recording channel.
```

```
  </p>
</equipment>
```

Example

```
<equipment>
  <p>8-track analogue transfer mixed down to 19 cm/sec audio
    tape for cassette mastering
  </p>
</equipment>
```

<equiv/> (equivalent) specifies a component which is considered equivalent to the parent element, either by co-reference, or by external link.

Module tagdocs — *22. Documentation Elements*

Attributes att.internetMedia (*@mimeType*)

@name names the underlying concept of which the parent is a representation

Status Optional

Datatype `data.name`

Values any name

@uri (uniform resource identifier) references the underlying concept of which the parent is a representation by means of some external identifier

Status Optional

Datatype `data.pointer`

Values a URI

@filter references an external script which contains a method to transform instances of this element to canonical TEI

Status Optional

Datatype `data.pointer`

Values a URI

Used by model.glossLike

May contain Empty element

Declaration

```
element equiv
{
   att.global.attributes,
   att.internetMedia.attributes,
   attribute name { data.name }?,
   attribute uri { data.pointer }?,
   attribute filter { data.pointer }?,
   empty
}
```

Example The following example declares that the <bo> element is conceptually equivalent to the markup construct **<hi rend='bold'>**, and that an external definition of this concept is available from the URI indicated

```
<elementSpec ident="hi" mode="change">
  <equiv name="BOLD"/>
  <desc>bold typography</desc>
  <attList>
    <attDef ident="rend">
      <valList>
        <valItem ident="bold"/>
      </valList>
    </attDef>
  </attList>
</elementSpec>
<elementSpec ident="bo" mode="add">
  <equiv name="BOLD" uri="http://www.typesrus.com/bold"/>
</elementSpec>
```

Note The **mimeType** attribute should be used to supply the MIME media type of the filter script specified by the **filter** attribute.

<etym> (etymology) encloses the etymological information in a dictionary entry.

Module dictionaries — *9. Dictionaries*

Attributes att.lexicographic (*@expand*, *@norm*, *@split*, *@value*, *@orig*, *@location*, *@mergedIn*, *@opt*)

Used by model.entryPart.top model.entryPart

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address bibl biblStruct binaryObject cb choice cit corr date del desc distinct email emph expan foreign gap gloss graphic hi index label lb list listBibl measure measureGrp mentioned milestone name note num orig pb ptr q quote ref reg rs said sic soCalled stage term time title unclear

dictionaries: case def gen gram iType lang lbl mood number oRef oVar pRef pVar per tns usg xr

drama: camera caption castList move sound tech view

figures: figure formula table

gaiji: g

header: biblFull

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material msDesc origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName listNym listOrg listPerson listPlace nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att classSpec code eg egXML elementSpec gi ident listRef macroSpec moduleRef moduleSpec specDesc specGrp specGrpRef specList tag val

textcrit: app listWit witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element etym
{
   att.global.attributes,
   att.lexicographic.attributes,
   (
      text
    | model.gLike    | model.phrase    | model.inter    | usg    | lbl    | def    | model.morphLike    | xr
}
```

Example

```
<entry>
  <form>
    <orth>publish</orth> ... </form>
  <etym>
    <lang>ME.</lang>
    <mentioned>publisshen</mentioned>, <lang>F.</lang>
    <mentioned>publier</mentioned>, <lang>L.</lang>
    <mentioned>publicare, publicatum</mentioned>. <xr>See <ref>public</ref>; cf. 2d
<ref>-ish</ref>.</xr>
  </etym>
</entry> (From: Webster's Second International)
```

Note May contain character data mixed with any other elements defined in the dictionary tag set.There is no consensus on the internal structure of etymologies, or even on whether such a structure is appropriate. The <etym> element accordingly simply contains prose, within which names of languages, cited words, or parts of words, glosses, and examples will typically be prominent. The tagging of such internal objects is optional.

<event> contains data relating to any kind of significant event associated with a person, place, or organization.

Module namesdates — *13. Names, Dates, People, and Places*

Attributes att.datableatt.datable.w3c (*@period*, *@when*, *@notBefore*, *@notAfter*, *@from*, *@to*) att.datable.iso (*@when-iso*, *@notBefore-iso*, *@notAfter-iso*, *@from-iso*, *@to-iso*) att.editLike (*@cert*, *@resp*, *@evidence*, *@source*, *@precision*) att.typed (*@type*, *@subtype*) att.naming (*@key*, *@ref*, *@nymRef*)

@where indicates the location of an event by pointing to a <place> element

Status Optional

Datatype `data.pointer`

Values any valid URI

Used by event model.persEventLike model.placeEventLike

May contain

core: bibl biblStruct desc head label note p

header: biblFull

linking: ab

msdescription: msDesc

namesdates: event

textcrit: witDetail

Declaration

```
element event
{
   att.global.attributes,
   att.datable.w3c.attributes,
   att.datable.iso.attributes,
   att.editLike.attributes,
   att.typed.attributes,
   att.naming.attributes,
   attribute where { data.pointer }?,
   (
      model.headLike*,
      ( ( model.pLike+ ) | ( model.labelLike+ ) ),
      ( model.noteLike | model.biblLike )*,
      event*
   )
}
```

Example

```
<person>
  <event  type="mat"  when="1972-10-12">
    <label>matriculation</label>
  </event>
  <event  type="grad"  when="1975-06-23">
    <label>graduation</label>
  </event>
</person>
```

<ex> (editorial expansion) contains a sequence of letters added by an editor or transcriber when expanding an abbreviation.

Module transcr — *11. Representation of Primary Sources*

Attributes att.editLike (*@cert*, *@resp*, *@evidence*, *@source*, *@precision*)

Used by model.pPart.editorial model.choicePart

May contain

gaiji: g

Declaration

```
element ex { att.global.attributes, att.editLike.attributes, macro.xtext }
```

Example

```
The address is Southmoor <choice>
  <expan>R<ex>oa</ex>d</expan>
  <abbr>Rd</abbr>
</choice>
```

<exemplum> groups an example demonstrating the use of an element along with optional paragraphs of commentary.

Module tagdocs — *22. Documentation Elements*

Attributes att.typed (*@type*, *@subtype*) att.translatable (*@version*)

Used by attDef classSpec elementSpec macroSpec moduleSpec

May contain

core: p

linking: ab

tagdocs: eg egXML

Declaration

```
element exemplum
{
   att.global.attributes,
   att.typed.attributes,
   att.translatable.attributes,
   ( model.pLike*, ( egXML | eg ), model.pLike* )
}
```

Example

```
<exemplum>
  <p>The <gi>name</gi> element can be used for both
     personal names and place names:</p>
  <eg><![CDATA[
     <q>My dear <name type="person">Mr. Bennet</name>,</q>
     said his lady to him one day, <q>have you heard that
     <name type="place">Netherfield Park</name> is let
     at last?</q>]]></eg>
  <p>As shown above, the <att>type</att> attribute may be used
     to distinguish the one from the other.</p>
</exemplum>
```

Note that an explicit end-tag must be supplied for the paragraph immediately preceding the <eg> element within an <exemplum>, to prevent the <eg> from being mistaken for part of the paragraph.

<expan> (expansion) contains the expansion of an abbreviation.

Module core — *3. Elements Available in All TEI Documents*

Attributes att.editLike (*@cert*, *@resp*, *@evidence*, *@source*, *@precision*)

Used by model.pPart.editorial model.choicePart

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address binaryObject cb choice corr date del distinct email emph expan foreign gap gloss graphic hi index lb measure measureGrp mentioned milestone name note num orig pb ptr ref reg rs sic soCalled term time title unclear

dictionaries: lang oRef oVar pRef pVar

figures: formula

gaiji: g

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident specDesc specList tag val

textcrit: app witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element expan
{
   att.global.attributes,
   att.editLike.attributes,
   macro.phraseSeq}
```

Example

```
The address is Southmoor <choice>
  <expan>Road</expan>
  <abbr>Rd</abbr>
</choice>
```

Note The content of this element should usually be a complete word or phrase. The <ex> element provided by the transcr module may be used to mark up sequences of letters supplied within such an expansion.

<explicit> contains the explicit of a manuscript item, that is, the closing words of the text proper, exclusive of any rubric or colophon which might follow it.

Module msdescription — *10. Manuscript Description*

Attributes att.typed (*@type*, *@subtype*) att.msExcerpt (*@defective*)

Used by msItemStruct model.msItemPart

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address binaryObject cb choice corr date del distinct email emph expan foreign gap gloss graphic hi index lb measure measureGrp mentioned milestone name note num orig pb ptr ref reg rs sic soCalled term time title unclear

dictionaries: lang oRef oVar pRef pVar

figures: formula

gaiji: g

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident specDesc specList tag val

textcrit: app witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element explicit
{
   att.global.attributes,
   att.typed.attributes,
   att.msExcerpt.attributes,
   macro.phraseSeq}
```

Example

```
<explicit>sed libera nos a malo.</explicit>
<rubric>Hic explicit oratio qui dicitur dominica.</rubric>
<explicit  type="defective">ex materia quasi et forma sibi
proporti<gap/>
</explicit>
<explicit  type="reverse">saued be shulle that doome of day the at
</explicit>
```

<extent> describes the approximate size of a text as stored on some carrier medium, whether digital or non-digital, specified in any convenient units.

Module header — *2. The TEI Header*

Attributes Global attributes only

Used by biblFull fileDesc monogr supportDesc model.biblPart

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address binaryObject cb choice corr date del distinct email emph expan foreign gap gloss graphic hi index lb measure measureGrp mentioned milestone name note num orig pb ptr ref reg rs sic soCalled term time title unclear

dictionaries: lang oRef oVar pRef pVar

figures: formula

gaiji: g

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident specDesc specList tag val

textcrit: app witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element extent { att.global.attributes, macro.phraseSeq }
```

Example

```
<extent>3200 sentences</extent>
<extent>between 10 and 20 Mb</extent>
<extent>ten 3.5 inch high density diskettes</extent>
```

<f> (feature) represents a feature value specification, that is, the association of a name with a value of any of several different types.

Module iso-fs — *18. Feature Structures*

Attributes In addition to global attributes

@name provides a name for the feature.

Status **Required**

Datatype `data.name`

Values Any name.

@fVal (feature value) references any element which can be used to represent the value of a feature.

Status Optional

Datatype `data.pointer`

Values the identifier of an element representing a feature value

Note If this attribute is supplied as well as content, the value referenced is to be unified with that contained.

Used by bicond cond fLib fs if

May contain

iso-fs: binary default fs numeric string symbol vAlt vColl vLabel vMerge vNot

Declaration

```
element f
{
   att.global.attributes,
   attribute name { data.name },
   attribute fVal { data.pointer }?,
   model.featureVal*
}
```

Example

```
<f  name="gender">
  <symbol  value="feminine"/>
</f>
```

Note If the element is empty then a value must be supplied for the **fVal** attribute.

<fDecl> (feature declaration) declares a single feature, specifying its name, organization, range of allowed values, and optionally its default value.

Module iso-fs — *18. Feature Structures*

Attributes In addition to global attributes

@name indicates the name of the feature being declared; matches the **name** attribute of <f> elements in the text.

Status **Required**

Datatype `data.name`

Values any string of characters

@optional indicates whether or not the value of this feature may be present.

Status Optional

Datatype `xsd:boolean`

Note If a feature is marked as optional, it is possible for it to be omitted from a feature structure. If an obligatory feature is omitted, then it is understood to have a default value, either explicitly declared, or, if no default is supplied, the special value **any**. If an optional feature is omitted, then it is understood to be missing and any possible value (including the default) is ignored.

Used by fsDecl

May contain

iso-fs: fDescr vDefault vRange

Declaration

```
element fDecl
{
   att.global.attributes,
   attribute name { data.name },
   attribute optional { xsd:boolean }?,
   ( fDescr?, vRange, vDefault? )
}
```

Example

```
<fDecl name="INV">
  <fDescr>inverted sentence</fDescr>
  <vRange>
    <vAlt>
      <binary value="true"/>
      <binary value="false"/>
    </vAlt>
  </vRange>
  <vDefault>
    <binary value="false"/>
  </vDefault>
</fDecl>
```

<fDescr> (feature description (in FSD)) describes in prose what is represented by the feature being declared and its values.

Module iso-fs — *18. Feature Structures*

Attributes Global attributes only

Used by fDecl

May contain

core: abbr address bibl biblStruct choice cit date desc distinct email emph expan foreign gloss label list listBibl measure measureGrp mentioned name num ptr q quote ref rs said soCalled stage term time title

dictionaries: lang

drama: camera caption castList move sound tech view

figures: figure table

header: biblFull

msdescription: catchwords depth dimensions height heraldry locus material msDesc origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName listNym listOrg listPerson listPlace nameLink offset orgName persName placeName region roleName settlement state surname

tagdocs: att classSpec code eg egXML elementSpec gi ident listRef macroSpec moduleRef moduleSpec specGrp specGrpRef tag val

textcrit: listWit

transcr: am ex handShift subst

Declaration

```
element fDescr { att.global.attributes, macro.limitedContent }
```

Example

```
<fDecl name="INV">
  <fDescr>inverted sentence</fDescr>
  <vRange>
    <vAlt>
      <binary value="true"/>
```

```
      <binary value="false"/>
    </vAlt>
  </vRange>
  <vDefault>
    <binary value="false"/>
  </vDefault>
</fDecl>
```

Note May contain character data, phrase-level elements, and inter-level elements.

<fLib> (feature library) assembles a library of feature elements.

Module iso-fs — *18. Feature Structures*

Attributes Global attributes only

Used by model.global.meta

May contain

iso-fs: f

Declaration

```
element fLib { att.global.attributes, f+ }
```

Example

```
<fLib n="agreement features">
  <f xml:id="pers1" name="person">
    <symbol value="first"/>
  </f>
  <f xml:id="pers2" name="person">
    <symbol value="second"/>
  </f>
<!-- ... -->
  <f xml:id="nums" name="number">
    <symbol value="singular"/>
  </f>
  <f xml:id="nump" name="number">
    <symbol value="plural"/>
  </f>
<!-- ... -->
</fLib>
```

Note The global **n** attribute may be used to supply an informal name to categorise the library's contents.

<facsimile> contains a representation of some written source in the form of a set of images rather than as transcribed or encoded text.

Module transcr — *11. Representation of Primary Sources*

Attributes att.declaring (*@decls*)

Used by model.resourceLike

May contain

core: binaryObject graphic

figures: formula

textstructure: back front

transcr: surface

Declaration

```
element facsimile
{
   att.global.attributes,
   att.declaring.attributes,
   ( front?, ( model.graphicLike | surface )+, back? )
}
```

Example

```
<facsimile>
  <graphic  url="page1.png"/>
  <surface>
    <graphic  url="page2-highRes.png"/>
    <graphic  url="page2-lowRes.png"/>
  </surface>
  <graphic  url="page3.png"/>
  <graphic  url="page4.png"/>
</facsimile>
```

Example

```
<facsimile>
  <surface
     ulx="0"
     uly="0"
     lrx="200"
     lry="300">
    <graphic  url="Bovelles-49r.png"/>
  </surface>
</facsimile>
```

<factuality> describes the extent to which the text may be regarded as imaginative or non-imaginative, that is, as describing a fictional or a non-fictional world.

Module corpus — *15. Language Corpora*

Attributes In addition to global attributes

@type categorizes the factuality of the text.

Status Optional

Legal values are: **fiction** the text is to be regarded as entirely imaginative

fact the text is to be regarded as entirely informative or factual

mixed the text contains a mixture of fact and fiction

inapplicable the fiction/fact distinction is not regarded as helpful or appropriate to this text

Used by model.textDescPart

May contain

analysis: interp interpGrp span spanGrp

certainty: certainty respons

core: abbr address cb choice date distinct email emph expan foreign gap gloss index lb measure measureGrp mentioned milestone name note num pb ptr ref rs soCalled term time title

dictionaries: lang

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident tag val

textcrit: witDetail

transcr: addSpan am damageSpan delSpan ex fw handShift space subst

Declaration

```
element factuality
{
   att.global.attributes,
   attribute type { "fiction" | "fact" | "mixed" | "inapplicable" }?,
   macro.phraseSeq.limited}
```

Example

```
<factuality type="fiction"/>
```

Example

```
<factuality type="mixed">contains a mixture of gossip and
speculation about real people and events</factuality>
```

Note Usually empty, unless some further clarification of the type attribute is needed, in which case it may contain running proseFor many literary texts, a simple binary opposition between 'fiction' and 'fact' is naïve in the extreme; this parameter is not intended for purposes of subtle literary analysis, but as a simple means of characterising the claimed fictiveness of a given text. No claim is made that works characterised as 'fact' are in any sense 'true'.

<faith> specifies the faith, religion, or belief set of a person.

Module namesdates — *13. Names, Dates, People, and Places*

Attributes att.editLike (*@cert*, *@resp*, *@evidence*, *@source*, *@precision*) att.datableatt.datable.w3c (*@period*, *@when*, *@notBefore*, *@notAfter*, *@from*, *@to*) att.datable.iso (*@when-iso*, *@notBefore-iso*, *@notAfter-iso*, *@from-iso*, *@to-iso*)

Used by model.persTraitLike

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address binaryObject cb choice corr date del distinct email emph expan foreign gap gloss graphic hi index lb measure measureGrp mentioned milestone name note num orig pb ptr ref reg rs sic soCalled term time title unclear

dictionaries: lang oRef oVar pRef pVar

figures: formula

gaiji: g

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident specDesc specList tag val

textcrit: app witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element faith
{
   att.global.attributes,
   att.editLike.attributes,
   att.datable.w3c.attributes,
   att.datable.iso.attributes,
   macro.phraseSeq}
```

Example

```
<faith>protestant</faith>
```

<figDesc>

<figDesc> (description of figure) contains a brief prose description of the appearance or content of a graphic figure, for use when documenting an image without displaying it.

Module figures — *14. Tables, Formulæ, and Graphics*

Attributes Global attributes only

Used by figure

May contain

core: abbr address bibl biblStruct choice cit date desc distinct email emph expan foreign gloss label list listBibl measure measureGrp mentioned name num ptr q quote ref rs said soCalled stage term time title

dictionaries: lang

drama: camera caption castList move sound tech view

figures: figure table

header: biblFull

msdescription: catchwords depth dimensions height heraldry locus material msDesc origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName listNym listOrg listPerson listPlace nameLink offset orgName persName placeName region roleName settlement state surname

tagdocs: att classSpec code eg egXML elementSpec gi ident listRef macroSpec moduleRef moduleSpec specGrp specGrpRef tag val

textcrit: listWit

transcr: am ex handShift subst

Declaration

```
element figDesc { att.global.attributes, macro.limitedContent }
```

Example

```
<figure>
  <graphic  url="emblem1.png"/>
  <head>Emblemi d'Amore</head>
  <figDesc>A pair of naked winged cupids, each holding a
     flaming torch, in a rural setting.</figDesc>
</figure>
```

Note This element is intended for use as an alternative to the content of its parent <figure> element; for example, to display when the image is required but the equipment in use cannot display graphic images. It may also be used for indexing or documentary purposes.

<figure> groups elements representing or containing graphic information such as an illustration or figure.

Module figures — *14. Tables, Formulæ, and Graphics*

Attributes att.placement (*@place*)

Used by figure model.inter model.titlepagePart

May contain

analysis: interp interpGrp span spanGrp

certainty: certainty respons

core: binaryObject cb gap graphic head index lb milestone note p pb

figures: figDesc figure formula

iso-fs: fLib fs fvLib

linking: ab alt altGrp anchor join joinGrp link linkGrp timeline

spoken: incident kinesic pause shift vocal writing

tagdocs: eg egXML

textcrit: witDetail

textstructure: floatingText

transcr: addSpan damageSpan delSpan fw space

Declaration

```
element figure
{
   att.global.attributes,
   att.placement.attributes,
   (
      model.headLike    | model.pLike    | figDesc    | model.graphicLike    | model.egLike    | floatingText
}
```

Example

```
<figure>
  <head>Figure One: The View from the Bridge</head>
  <figDesc>A Whistleresque view showing four
     or five sailing boats in the foreground, and a
     series of buoys strung out between them.</figDesc>
  <graphic  url="http://www.example.org/fig1.png"  scale="0.5"/>
</figure>
```

<fileDesc> (file description) contains a full bibliographic description of an electronic file.

Module header — *2. The TEI Header*

Attributes Global attributes only

Used by teiHeader

May contain

header: editionStmt extent notesStmt publicationStmt seriesStmt sourceDesc titleStmt

Declaration

```
element fileDesc
{
   att.global.attributes,
   (
      (
         titleStmt,
         editionStmt?,
         extent?,
         publicationStmt,
         seriesStmt?,
         notesStmt?
      ),
      sourceDesc+
   )
}
```

Example

```
<fileDesc>
  <titleStmt>
    <title>The shortest possible TEI document</title>
  </titleStmt>
  <publicationStmt>
    <p>Distributed as part of TEI P5</p>
  </publicationStmt>
```

```
  <sourceDesc>
    <p>No print source exists: this is an original
        digital text</p>
  </sourceDesc>
</fileDesc>
```

Note The major source of information for those seeking to create a catalogue entry or bibliographic citation for an electronic file. As such, it provides a title and statements of responsibility together with details of the publication or distribution of the file, of any series to which it belongs, and detailed bibliographic notes for matters not addressed elsewhere in the header. It also contains a full bibliographic description for the source or sources from which the electronic text was derived.

<filiation> contains information concerning the manuscript's filiation, i.e. its relationship to other surviving manuscripts of the same text, its protographs, antigraphs and apographs.

Module msdescription — *10. Manuscript Description*

Attributes att.typed (*@type*, *@subtype*)

Used by msItemStruct model.msItemPart

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address bibl biblStruct binaryObject cb choice cit corr date del desc distinct email emph expan foreign gap gloss graphic hi index l label lb lg list listBibl measure measureGrp mentioned milestone name note num orig p pb ptr q quote ref reg rs said sic soCalled sp stage term time title unclear

dictionaries: lang oRef oVar pRef pVar

drama: camera caption castList move sound tech view

figures: figure formula table

gaiji: g

header: biblFull

iso-fs: fLib fs fvLib

linking: ab alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material msDesc origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName listNym listOrg listPerson listPlace nameLink offset orgName persName placeName region roleName settlement state surname

nets: eTree forest forestGrp graph tree

spoken: incident kinesic pause shift u vocal writing

tagdocs: att classSpec code eg egXML elementSpec gi ident listRef macroSpec moduleRef moduleSpec schemaSpec specDesc specGrp specGrpRef specList tag val

textcrit: app listWit witDetail

textstructure: floatingText

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element filiation
{
   att.global.attributes,
   att.typed.attributes,
   macro.specialPara}
```

Example

```
<msContents>
  <msItem>
    <title>Beljakovski sbornik</title>
    <filiation type="protograph">Bulgarian</filiation>
    <filiation type="antigraph">Middle Bulgarian</filiation>
    <filiation type="apograph">
      <ref target="#DN17">Dujchev N 17</ref>
    </filiation>
  </msItem>
</msContents>
<!-- ... -->
<msDesc xml:id="DN17">
<!-- ... -->
</msDesc>
```

In this example, the reference to 'Dujchev N17' includes a link to some other manuscript description which has the identifier `DN17`.

Example

```
<msItem>
  <title>Guan-ben</title>
  <filiation>
    <p>The "Guan-ben" was widely current among mathematicians in the
        Qing dynasty, and "Zhao Qimei version" was also read. It is
        therefore difficult to know the correct filiation path to follow.
        The study of this era is much indebted to Li Di. We explain the
        outline of his conclusion here. Kong Guangsen
        (1752-1786)(17) was from the same town as Dai Zhen, so he obtained
        "Guan-ben" from him and studied it(18). Li Huang (d. 1811)
        (19) took part in editing Si Ku Quan Shu, so he must have had
        "Guan-ben". Then Zhang Dunren (1754-1834) obtained this version,
        and studied "Da Yan Zong Shu Shu" (The General Dayan
        Computation). He wrote Jiu Yi Suan Shu (Mathematics
        Searching for One, 1803) based on this version of Shu Xue Jiu
        Zhang (20).</p>
    <p>One of the most important persons in restoring our knowledge
        concerning the filiation of these books was Li Rui (1768(21)
        -1817)(see his biography). ... only two volumes remain of this
        manuscript, as far as chapter 6 (chapter 3 part 2) p.13, that is,
        question 2 of "Huan Tian San Ji" (square of three loops),
        which later has been lost.</p>
  </filiation>
```

```
</msItem>
<!--http://www2.nkfust.edu.tw/~jochi/ed1.htm-->
```

<finalRubric> contains the string of words that denotes the end of a text division, often with an assertion as to its author and title, usually set off from the text itself by red ink, by a different size or type of script, or by some other such visual device.

Module msdescription — *10. Manuscript Description*

Attributes att.typed (*@type*, *@subtype*)

Used by msItemStruct model.msItemPart

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address binaryObject cb choice corr date del distinct email emph expan foreign gap gloss graphic hi index lb measure measureGrp mentioned milestone name note num orig pb ptr ref reg rs sic soCalled term time title unclear

dictionaries: lang oRef oVar pRef pVar

figures: formula

gaiji: g

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident specDesc specList tag val

textcrit: app witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element finalRubric
{
   att.global.attributes,
   att.typed.attributes,
   macro.phraseSeq}
```

Example

```
<finalRubric>Explicit le romans de la Rose ou l'art
d'amours est toute enclose.</finalRubric>
<finalRubric>ok lúkv ver þar Brennu-Nials savgv</finalRubric>
```

<floatingText> contains a single text of any kind, whether unitary or composite, which interrupts the text containing it at any point and after which the surrounding text resumes.

Module textstructure — *4. Default Text Structure*

Attributes att.declaring (*@decls*) att.typed (*@type*, *@subtype*)

Used by figure model.divPart

May contain

analysis: interp interpGrp span spanGrp

certainty: certainty respons

core: cb gap index lb milestone note pb

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp timeline

spoken: incident kinesic pause shift vocal writing

textcrit: witDetail

textstructure: back body front group

transcr: addSpan damageSpan delSpan fw space

Declaration

```
element floatingText
{
   att.global.attributes,
   att.declaring.attributes,
   att.typed.attributes,
   (
      model.global*,
      ( front, model.global* )?,
      ( body | group ),
      model.global*,
      ( back, model.global* )?
   )
}
```

Example

```
<TEI>
  <teiHeader/>
  <text>
    <body>
      <div type="scene">
        <sp>
          <p>Hush, the players begin...</p>
        </sp>
        <floatingText type="pwp">
          <body>
            <div type="act">
              <sp>
                <l>In Athens our tale takes place ....</l>
              </sp>
<!-- ... rest of nested act here -->
            </div>
          </body>
```

```
        </floatingText>
        <sp>
          <p>Now that the play is finished ...</p>
        </sp>
      </div>
    </body>
  </text>
</TEI>
```

Note A floating text has the same content as any other and may thus be interrupted by another floating text, or contain a group of tesselated texts

---

<floruit> contains information about a person's period of activity.

Module namesdates — *13. Names, Dates, People, and Places*

Attributes att.datableatt.datable.w3c (*@period*, *@when*, *@notBefore*, *@notAfter*, *@from*, *@to*) att.datable.iso (*@when-iso*, *@notBefore-iso*, *@notAfter-iso*, *@from-iso*, *@to-iso*) att.editLike (*@cert*, *@resp*, *@evidence*, *@source*, *@precision*)

Used by model.persStateLike

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address binaryObject cb choice corr date del distinct email emph expan foreign gap gloss graphic hi index lb measure measureGrp mentioned milestone name note num orig pb ptr ref reg rs sic soCalled term time title unclear

dictionaries: lang oRef oVar pRef pVar

figures: formula

gaiji: g

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident specDesc specList tag val

textcrit: app witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element floruit
{
   att.global.attributes,
```

```
  att.datable.w3c.attributes,
  att.datable.iso.attributes,
  att.editLike.attributes,
  macro.phraseSeq}
```

Example

```
<floruit notBefore="1066" notAfter="1100"/>
```

<foliation> describes the numbering system or systems used to count the leaves or pages in a codex.

Module msdescription — *10. Manuscript Description*

Attributes Global attributes only

Used by supportDesc

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address bibl biblStruct binaryObject cb choice cit corr date del desc distinct email emph expan foreign gap gloss graphic hi index l label lb lg list listBibl measure measureGrp mentioned milestone name note num orig p pb ptr q quote ref reg rs said sic soCalled sp stage term time title unclear

dictionaries: lang oRef oVar pRef pVar

drama: camera caption castList move sound tech view

figures: figure formula table

gaiji: g

header: biblFull

iso-fs: fLib fs fvLib

linking: ab alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material msDesc origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName listNym listOrg listPerson listPlace nameLink offset orgName persName placeName region roleName settlement state surname

nets: eTree forest forestGrp graph tree

spoken: incident kinesic pause shift u vocal writing

tagdocs: att classSpec code eg egXML elementSpec gi ident listRef macroSpec moduleRef moduleSpec schemaSpec specDesc specGrp specGrpRef specList tag val

textcrit: app listWit witDetail

textstructure: floatingText

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element foliation { att.global.attributes, macro.specialPara }
```

Example

```
<foliation>Contemporary foliation in red
roman numerals in the centre
of the outer margin.</foliation>
```

<foreign> (foreign) identifies a word or phrase as belonging to some language other than that of the surrounding text.

Module core — *3. Elements Available in All TEI Documents*

Attributes Global attributes only

Used by model.emphLike

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address binaryObject cb choice corr date del distinct email emph expan foreign gap gloss graphic hi index lb measure measureGrp mentioned milestone name note num orig pb ptr ref reg rs sic soCalled term time title unclear

dictionaries: lang oRef oVar pRef pVar

figures: formula

gaiji: g

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident specDesc specList tag val

textcrit: app witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element foreign { att.global.attributes, macro.phraseSeq }
```

Example

```
This is heathen Greek to you still?
Your <foreign  xml:lang="la">lapis philosophicus</foreign>?
```

Source: [108]

Note The global **xml:lang** attribute should be supplied for this element to identify the language of the word or phrase marked. As elsewhere, its value should be a language tag as defined in *vi.1*

*Language identification.*This element is intended for use only where no other element is available to mark the phrase or words concerned. The global **xml:lang** attribute should be used in preference to this element where it is intended to mark the language of the whole of some text element.The <distinct> element may be used to identify phrases belonging to sublanguages or registers not generally regarded as true languages.

<forename> contains a forename, given or baptismal name.

Module namesdates — *13. Names, Dates, People, and Places*

Attributes att.personal (*@full*, *@sort*) att.naming (*@key*, *@ref*, *@nymRef*) att.typed (*@type*, *@subtype*)

Used by model.persNamePart

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address binaryObject cb choice corr date del distinct email emph expan foreign gap gloss graphic hi index lb measure measureGrp mentioned milestone name note num orig pb ptr ref reg rs sic soCalled term time title unclear

dictionaries: lang oRef oVar pRef pVar

figures: formula

gaiji: g

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident specDesc specList tag val

textcrit: app witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element forename
{
   att.global.attributes,
   att.personal.attributes,
   att.naming.attributes,
   att.typed.attributes,
   macro.phraseSeq}
```

Example

```
<persName>
  <roleName>Ex-President</roleName>
  <forename>George</forename>
  <surname>Bush</surname>
</persName>
```

<forest> provides for groups of rooted trees.

Module nets — *19. Graphs, Networks, and Trees*

Attributes In addition to global attributes

@type identifies the type of the forest.

Status Optional

Datatype `data.enumerated`

Values A character string.

Used by forestGrp model.divPart

May contain

nets: eTree tree triangle

Declaration

```
element forest
{
   att.global.attributes,
   attribute type { data.enumerated }?,
   ( tree | eTree | triangle )+
}
```

Example

```
<forest n="ex5" type="derivation-syntactic">
  <eTree n="Stage 1" xml:id="s1SBAR">
    <label>S'</label>
    <eTree xml:id="s1S">
      <label>S</label>
      <eTree xml:id="s1NP1">
        <label>NP</label>
        <eLeaf>
          <label>you</label>
        </eLeaf>
      </eTree>
      <eTree xml:id="s1VP">
        <label>VP</label>
        <eTree xml:id="s1V">
          <label>V</label>
          <eLeaf>
            <label>do</label>
          </eLeaf>
        </eTree>
      </eTree>
    </eTree>
  </eTree>
  <eTree n="Stage 2" xml:id="s2SBAR" corresp="#s1SBAR">
```

```
      <label>S'</label>
      <eTree xml:id="s2S" corresp="#s1S">
        <label>S</label>
        <eTree xml:id="s2NP1" copyOf="#s1NP1">
          <label>NP</label>
        </eTree>
        <eTree xml:id="s2VP" corresp="#s1VP">
          <label>VP</label>
          <eTree xml:id="s2V" copyOf="#s1V">
            <label>V</label>
          </eTree>
          <eTree xml:id="s2NP2" corresp="#s1NP2">
            <label>NP</label>
            <eLeaf corresp="#s1WH">
              <label>t</label>
            </eLeaf>
          </eTree>
        </eTree>
      </eTree>
    </forest>
```

Note One or more trees, embedding trees, or underspecified embedding trees (triangles).

<forestGrp> (forest group) provides for groups of forests.

Module nets — *19. Graphs, Networks, and Trees*

Attributes In addition to global attributes

@type identifies the type of the forest group.

Status Optional

Datatype `data.enumerated`

Values A character string.

Used by model.divPart

May contain

nets: forest

Declaration

```
element forestGrp
{
   att.global.attributes,
   attribute type { data.enumerated }?,
   forest+
}
```

Note One or more forests.

<form> (form information group) groups all the information on the written and spoken forms of one headword.

Module dictionaries — *9. Dictionaries*

Attributes att.lexicographic (*@expand*, *@norm*, *@split*, *@value*, *@orig*, *@location*, *@mergedIn*, *@opt*)

@type classifies form as simple, compound, etc.

Status Optional

Datatype `data.enumerated`

Suggested values include: **simple** single free lexical item

lemma the headword itself

variant a variant form

compound word formed from simple lexical items

derivative word derived from headword

inflected word in other than usual dictionary form

phrase multiple-word lexical item

Used by superEntry model.entryPart.top model.entryPart model.formPart

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address bibl biblStruct binaryObject cb choice cit corr date del desc distinct email emph expan foreign gap gloss graphic hi index label lb list listBibl measure measureGrp mentioned milestone name note num orig pb ptr q quote ref reg rs said sic soCalled stage term time title unclear

dictionaries: case colloc form gen gram gramGrp hyph iType lang lbl mood number oRef oVar orth pRef pVar per pos pron subc syll tns usg

drama: camera caption castList move sound tech view

figures: figure formula table

gaiji: g

header: biblFull

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material msDesc origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName listNym listOrg listPerson listPlace nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att classSpec code eg egXML elementSpec gi ident listRef macroSpec moduleRef moduleSpec specDesc specGrp specGrpRef specList tag val

textcrit: app listWit witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element form
{
   att.global.attributes,
   att.lexicographic.attributes,
```

```
   attribute type
   {
      "simple"
    | "lemma"
    | "variant"
    | "compound"
    | "derivative"
    | "inflected"
    | "phrase"
    | xsd:Name
   }?,
   (
      text
    | model.gLike    | model.phrase    | model.inter    | model.formPart    | model.global   )*
}
```

Example

```
<form>
  <orth>zaptié</orth>
  <orth>zaptyé</orth>
</form>
```

(from TLFi)

<formula> contains a mathematical or other formula.

Module figures — *14. Tables, Formulæ, and Graphics*

Attributes In addition to global attributes

@notation supplies the name of a previously defined notation used for the content of the element.

Status Optional

Datatype data.code

Values The name of a formal notation previously declared in the document type declaration.

Used by model.graphicLike

May contain

core: binaryObject graphic

figures: formula

Declaration

```
element formula
{
   att.global.attributes,
   attribute notation { data.code }?,
   ( text | model.graphicLike )*
}
```

Example

```
<formula notation="TeX">$e=mc^2$</formula>
```

<front> (front matter) contains any prefatory matter (headers, title page, prefaces, dedications, etc.) found at the start of a document, before the main body.

Module textstructure — *4. Default Text Structure*

Attributes att.declaring (*@decls*)

Used by facsimile floatingText text

May contain

analysis: interp interpGrp span spanGrp

certainty: certainty respons

core: cb divGen gap head index lb milestone note pb

drama: castList epilogue performance prologue set

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp timeline

spoken: incident kinesic pause shift vocal writing

textcrit: witDetail

textstructure: argument byline closer div div1 docAuthor docDate docEdition docImprint docTitle epigraph postscript signed titlePage titlePart trailer

transcr: addSpan damageSpan delSpan fw space

Declaration

```
element front
{
   att.global.attributes,
   att.declaring.attributes,
   (
      ( model.frontPart | model.pLike.front | model.global )*,
      (
         (
            (
               ( model.div1Like ),
               ( model.frontPart | model.div1Like | model.global )*
            )
          | (
               ( model.divLike ),
               ( model.frontPart | model.divLike | model.global )*
            )
         )?
      ),
      ( ( ( model.divBottomPart ), ( model.divBottomPart | model.global )* )? )
   )
}
```

Example

```
<front>
  <epigraph>
    <quote>Nam Sibyllam quidem Cumis ego ipse oculis meis
```

```
        vidi in ampulla pendere, et cum illi pueri dicerent:
    <q xml:lang="grc">Sibylla ti weleis</q>; respondebat
        illa: <q xml:lang="grc">apowanein welo.</q>
    </quote>
  </epigraph>
  <div type="dedication">
    <p>For Ezra Pound <q xml:lang="it">il miglior fabbro.</q>
    </p>
  </div>
</front>
```

Example

```
<front>
  <div type="dedication">
    <p>To our three selves</p>
  </div>
  <div type="preface">
    <head>Author's Note</head>
    <p>All the characters in this book are purely imaginary, and if the
        author has used names that may suggest a reference to living persons
        she has done so inadvertently.
        ...</p>
  </div>
</front>
```

<fs> (feature structure) represents a feature structure, that is, a collection of feature-value pairs organized as a structural unit.

Module iso-fs — *18. Feature Structures*

Attributes In addition to global attributes

@type specifies the type of the feature structure.

Status Required when applicable

Datatype `data.enumerated`

Values Character string, e.g. word structure.

@feats (features) references the feature-value specifications making up this feature structure.

Status Optional

Datatype 1–∞ occurrences of `data.pointer` separated by whitespace

Values one or more identifiers of <f> elements.

Note May be used either instead of having features as content, or in addition. In the latter case, the features referenced and contained are unified.

Used by bicond cond if vColl model.featureVal.complex model.global.meta

May contain

iso-fs: f

Declaration

```
element fs
{
   att.global.attributes,
   attribute type { data.enumerated }?,
```

```
   attribute feats { list { data.pointer+ } }?,
   f*
}
```

Example

```
<fs type="agreement_structure">
  <f name="person">
    <symbol value="third"/>
  </f>
  <f name="number">
    <symbol value="singular"/>
  </f>
</fs>
```

<fsConstraints> (feature-structure constraints) specifies constraints on the content of valid feature structures.

Module iso-fs — *18. Feature Structures*

Attributes Global attributes only

Used by fsDecl

May contain

iso-fs: bicond cond

Declaration

```
element fsConstraints { att.global.attributes, ( cond | bicond )* }
```

Example

```
<fsConstraints>
  <cond>
    <fs>
<!-- ...-->
    </fs>
    <then/>
    <fs>
<!-- ... -->
    </fs>
  </cond>
  <bicond>
<!-- ... -->
  </bicond>
  <cond>
<!-- ... -->
  </cond>
</fsConstraints>
```

Note May contain a series of conditional or biconditional elements.

<fsDecl> (feature structure declaration) declares one type of feature structure.

Module iso-fs — *18. Feature Structures*

Attributes In addition to global attributes

@type gives a name for the type of feature structure being declared.

Status **Required**

Datatype `data.enumerated`

Values any convenient string of characters.

@baseTypes gives the name of one or more typed feature structures from which this type inherits feature specifications and constraints; if this type includes a feature specification with the same name as that of any of those specified by this attribute, or if more than one specification of the same name is inherited, then the set of possible values is defined by unification. Similarly, the set of constraints applicable is derived by combining those specified explicitly within this element with those implied by the **baseTypes** attribute. When no **baseTypes** attribute is specified, no feature specification or constraint is inherited.

Status Optional

Datatype 1–∞ occurrences of `data.name` separated by whitespace

Values one or more names as defined by the W3C XML Specification

Note Inheritance is defined here as a monotonous relation.The process of combining constraints may result in a contradiction, for example if two specifications for the same feature specify disjoint ranges of values, and at least one such specification is mandatory. In such a case, there is no valid representative for the type being defined.

Used by fsdDecl

May contain

iso-fs: fDecl fsConstraints fsDescr

Declaration

```
element fsDecl
{
   att.global.attributes,
   attribute type { data.enumerated },
   attribute baseTypes { list { data.name+ } }?,
   ( fsDescr?, fDecl+, fsConstraints? )
}
```

Example

```
<fsDecl  type="SomeName">
  <fsDescr>Describes what this type of fs represents</fsDescr>
  <fDecl  name="featureOne">
<!-- The declaration for featureOne -->
  </fDecl>
  <fDecl  name="featureTwo">
<!-- The declaration for featureTwo -->
  </fDecl>
  <fsConstraints>
<!-- The feature structure constraints go here -->
  </fsConstraints>
</fsDecl>
```

<fsDescr> (feature system description (in FSD)) describes in prose what is represented by the type of feature structure declared in the enclosing fsDecl.

Module iso-fs — *18. Feature Structures*

Attributes Global attributes only

Used by fsDecl

May contain

core: abbr address bibl biblStruct choice cit date desc distinct email emph expan foreign gloss label list listBibl measure measureGrp mentioned name num ptr q quote ref rs said soCalled stage term time title

dictionaries: lang

drama: camera caption castList move sound tech view

figures: figure table

header: biblFull

msdescription: catchwords depth dimensions height heraldry locus material msDesc origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName listNym listOrg listPerson listPlace nameLink offset orgName persName placeName region roleName settlement state surname

tagdocs: att classSpec code eg egXML elementSpec gi ident listRef macroSpec moduleRef moduleSpec specGrp specGrpRef tag val

textcrit: listWit

transcr: am ex handShift subst

Declaration

```
element fsDescr { att.global.attributes, macro.limitedContent }
```

Example

```
<fsDecl type="Agreement">
  <fsDescr>This type of feature structure encodes the features
     for subject-verb agreement in English</fsDescr>
  <fDecl name="PERS">
    <fDescr>person (first, second, or third)</fDescr>
<!-- ... -->
  </fDecl>
  <fDecl name="NUM">
    <fDescr>number (singular or plural)</fDescr>
<!-- ...-->
  </fDecl>
</fsDecl>
```

Note May contain character data, phrase-level elements, and inter-level elements.

<fsdDecl> (feature system declaration) provides a feature system declaration comprising one or more feature structure declarations or feature structure declaration links.

Module iso-fs — *18. Feature Structures*

Attributes Global attributes only

Used by model.encodingPart model.resourceLike

May contain

iso-fs: fsDecl fsLink

Declaration

```
element fsdDecl { att.global.attributes, ( fsDecl | fsLink )+ }
```

Example

```
<fsdDecl>
  <fsDecl type="GPSG">
<!-- ... -->
  </fsDecl>
  <fsDecl type="lex" xml:id="LX123">
<!-- ... -->
  </fsDecl>
  <fsLink type="entry" target="#LX123"/>
  <fsLink
     type="subentry"
     target="http://www.example.com/fsdLib.xml#LX123"/>
</fsdDecl>
```

<fsdLink/>

(feature structure declaration link) associates the name of a typed feature structure with a feature structure declaration for it.

Module iso-fs — *18. Feature Structures*

Attributes In addition to global attributes

@type identifies the type of feature structure to be documented; this will be the value of the **type** attribute on at least one feature structure.

Status **Required**

Datatype `data.enumerated`

Values any string of characters.

@target supplies a pointer to a feature structure declaration (<fsDecl>) element within the current document or elsewhere.

Status **Required**

Datatype `data.pointer`

Used by fsdDecl

May contain Empty element

Declaration

```
element fsdLink
{
   att.global.attributes,
   attribute type { data.enumerated },
   attribute target { data.pointer },
   empty
}
```

Example

```
<fsdLink
   type="subentry"
   target="http://www.example.com/fsdLib.xml#L1234"/>
```

<funder> (funding body) specifies the name of an individual, institution, or organization responsible for the funding of a project or text.

Module header — *2. The TEI Header*

Attributes Global attributes only

Used by titleStmt

May contain

analysis: interp interpGrp span spanGrp

certainty: certainty respons

core: abbr address cb choice date distinct email emph expan foreign gap gloss index lb measure measureGrp mentioned milestone name note num pb ptr ref rs soCalled term time title

dictionaries: lang

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident tag val

textcrit: witDetail

transcr: addSpan am damageSpan delSpan ex fw handShift space subst

Declaration

```
element funder { att.global.attributes, macro.phraseSeq.limited }
```

Example

```
<funder>The National Endowment for the Humanities, an independent
federal agency</funder>
<funder>Directorate General XIII of the Commission of the
European Communities</funder>
<funder>The Andrew W. Mellon Foundation</funder>
<funder>The Social Sciences and Humanities Research Council of Canada</funder>
```

Note Funders provide financial support for a project; they are distinct from sponsors, who provide intellectual support and authority.

<fvLib> (feature-value library) assembles a library of reusable feature value elements (including complete feature structures).

Module iso-fs — *18. Feature Structures*

Attributes Global attributes only

Used by model.global.meta

May contain

iso-fs: binary default fs numeric string symbol vAlt vColl vLabel vMerge vNot

Declaration

```
element fvLib { att.global.attributes, model.featureVal* }
```

Example

```
<fvLib n="symbolic values">
  <symbol xml:id="sfirst" value="first"/>
  <symbol xml:id="ssecond" value="second"/>
<!-- ... -->
  <symbol xml:id="ssing" value="singular"/>
  <symbol xml:id="splur" value="plural"/>
<!-- ... -->
</fvLib>
```

Note A feature value library may include any number of values of any kind, including multiple occurrences of identical values such as `<binary value="true"/>` or `default`. The only thing guaranteed unique in a feature value library is the set of labels used to identify the values.

<fw> (forme work) contains a running head (e.g. a header, footer), catchword, or similar material appearing on the current page.

Module transcr — *11. Representation of Primary Sources*

Attributes att.placement (*@place*)

@type classifies the material encoded according to some useful typology.

Status Recommended

Datatype `data.enumerated`

Sample values include: **header** a running title at the top of the page

footer a running title at the bottom of the page

pageNum (page number) a page number or foliation symbol

lineNum (line number) a line number, either of prose or poetry

sig (signature) a signature or gathering symbol

catch (catchword) a catch-word

Used by model.milestoneLike

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address binaryObject cb choice corr date del distinct email emph expan foreign gap gloss graphic hi index lb measure measureGrp mentioned milestone name note num orig pb ptr ref reg rs sic soCalled term time title unclear

dictionaries: lang oRef oVar pRef pVar

figures: formula

gaiji: g

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident specDesc specList tag val

textcrit: app witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element fw
{
   att.global.attributes,
   att.placement.attributes,
   attribute type { data.enumerated }?,
   macro.phraseSeq}
```

Example

```
<fw type="sig" place="bot">C3</fw>
```

Note Where running heads are consistent throughout a chapter or section, it is usually more convenient to relate them to the chapter or section, e.g. by use of the **rend** attribute. The <fw> element is intended for cases where the running head changes from page to page, or where details of page layout and the internal structure of the running heads are of paramount importance.

<g> (character or glyph) represents a non-standard character or glyph.

Module gaiji — *5. Representation of Non-standard Characters and Glyphs*

Attributes att.typed (*@type*, *@subtype*)

@ref points to a description of the character or glyph intended.

Status Optional

Datatype `data.pointer`

Values a pointer to some another element.

Used by model.gLike

May contain Character data only

Declaration

```
element g
{
```

```
    att.global.attributes,
    att.typed.attributes,
    attribute ref { data.pointer }?,
    text
}
```

Example

```
<g ref="#flig">fl</g>
```

This example points to a <glyph> element with the identifier `flig` like the following:

```
<glyph xml:id="flig">
<!--...-->
</glyph>
```

Note The name g is short for gaiji, which is the Japanese term for a non-standardized character or glyph.

<gap> indicates a point where material has been omitted in a transcription, whether for editorial reasons described in the TEI header, as part of sampling practice, or because the material is illegible, invisible, or inaudible.

Module core — *3. Elements Available in All TEI Documents*

Attributes att.durationatt.duration.w3c (*@dur*) att.duration.iso (*@dur-iso*) att.editLike (*@cert*, *@resp*, *@evidence*, *@source*, *@precision*)

@reason gives the reason for omission. Sample values include **sampling**, **illegible**, **inaudible**, **irrelevant**, **cancelled**, **illegible**.

Status Optional

Datatype 1–∞ occurrences of `data.word` separated by whitespace

Values any short indication of the reason for the omission.

@hand in the case of text omitted from the transcription because of deliberate deletion by an identifiable hand, signifies the hand which made the deletion.

Status Optional

Datatype `data.pointer`

Values must be one of the hand identifiers declared in the document header (see section *11.4.1. Document Hands*).

@agent In the case of text omitted because of damage, categorizes the cause of the damage, if it can be identified.

Status Optional

Datatype `data.enumerated`

Sample values include: **rubbing** damage results from rubbing of the leaf edges

mildew damage results from mildew on the leaf surface

smoke damage results from smoke

@extent indicates approximately how much text has been omitted from the transcription (using the unit specified on the **unit** attribute), either because of editorial policy or because a deletion, damage, or other cause has rendered transcription impossible.

Status Optional

Datatype `data.numeric`

Note It is possible, but not always necessary, to provide measurements precise to the millimeter or even to the printer's point. The degree of precision attempted will vary with the purpose of the encoding and the nature of the material.

@unit names the unit used for describing the extent of the gap

Status Optional

Datatype `data.enumerated`

Suggested values include: **lines** lines of text

chars (characters) characters of text

pages pages, i.e. one side of a leaf

cm (centimetres)

mm (millimetres)

in (inches)

Used by model.global.edit

May contain

core: desc gloss

tagdocs: altIdent equiv

Declaration

```
element gap
{
   att.global.attributes,
   att.duration.w3c.attributes,
   att.duration.iso.attributes,
   att.editLike.attributes,
   attribute reason { list { data.word+ } }?,
   attribute hand { data.pointer }?,
   attribute agent { data.enumerated }?,
   attribute extent { data.numeric }?,
   attribute unit
   {
      "lines" | "chars" | "pages" | "cm" | "mm" | "in" | xsd:Name
   }?,
   model.glossLike*
}
```

Example

```
<gap extent="4" unit="chars" reason="illegible"/>
```

Example

```
<gap extent="1" unit="essay" reason="sampling"/>
```

Note The <gap>, <unclear>, and <del> core tag elements may be closely allied in use with the <damage> and <supplied> elements, available when using the additional tagset for transcription of primary sources. See section *11.5.2. Use of the <gap>, <del>, <damage>, <unclear>, and <supplied> Elements in Combination* for discussion of which element is appropriate for which circumstance.

<gen> (gender) identifies the morphological gender of a lexical item, as given in the dictionary.

Module dictionaries — *9. Dictionaries*

Attributes att.lexicographic (*@expand*, *@norm*, *@split*, *@value*, *@orig*, *@location*, *@mergedIn*, *@opt*)

Used by model.entryPart model.morphLike

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address bibl biblStruct binaryObject cb choice cit corr date del desc distinct email emph expan foreign gap gloss graphic hi index label lb list listBibl measure measureGrp mentioned milestone name note num orig pb ptr q quote ref reg rs said sic soCalled stage term time title unclear

dictionaries: lang oRef oVar pRef pVar

drama: camera caption castList move sound tech view

figures: figure formula table

gaiji: g

header: biblFull

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material msDesc origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName listNym listOrg listPerson listPlace nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att classSpec code eg egXML elementSpec gi ident listRef macroSpec moduleRef moduleSpec specDesc specGrp specGrpRef specList tag val

textcrit: app listWit witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element gen
{
   att.global.attributes,
   att.lexicographic.attributes,
   macro.paraContent}
```

Example

```
<entry>
  <form>
    <orth>pamplemousse</orth>
  </form>
  <gramGrp>
    <pos>noun</pos>
    <gen>masculine</gen>
```

```
    </gramGrp>
  </entry>
```

Note May contain character data and phrase-level elements. Typical content will be *masculine*, *feminine*, *neuter* etc.This element is synonymous with **<gram type="gender">**.

<genName> (generational name component) contains a name component used to distinguish otherwise similar names on the basis of the relative ages or generations of the persons named.

Module namesdates — *13. Names, Dates, People, and Places*

Attributes att.personal (*@full*, *@sort*) att.naming (*@key*, *@ref*, *@nymRef*) att.typed (*@type*, *@subtype*)

Used by model.persNamePart

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address binaryObject cb choice corr date del distinct email emph expan foreign gap gloss graphic hi index lb measure measureGrp mentioned milestone name note num orig pb ptr ref reg rs sic soCalled term time title unclear

dictionaries: lang oRef oVar pRef pVar

figures: formula

gaiji: g

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident specDesc specList tag val

textcrit: app witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element genName
{
   att.global.attributes,
   att.personal.attributes,
   att.naming.attributes,
   att.typed.attributes,
   macro.phraseSeq}
```

Example

```
<persName>
  <forename>Charles</forename>
  <genName>II</genName>
</persName>
```

Example

```
<persName>
  <surname>Pitt</surname>
  <genName>the Younger</genName>
</persName>
```

<geo> (geographical coordinates) contains any expression of a set of geographic coordinates, representing a point, line, or area on the surface of the earth in some notation.

Module namesdates — *13. Names, Dates, People, and Places*

Attributes Global attributes only

Used by model.measureLike

May contain Character data only

Declaration `element geo { att.global.attributes, text }`

Example

```
<geo>41.687142 -74.870109</geo>
```

Note All uses of <geo> within a document are required to use the same coordinate system, which is that defined by a <geoDecl> element supplied in the TEI Header. If no such element is supplied, the assumption is that the content of each <geo> element will be a pair of numbers separated by whitespace, to be interpreted as latitude followed by longitude according to the World Geodetic System.

<geoDecl> (geographic coordinates declaration) documents the notation and the datum used for geographic coordinates expressed as content of the <geo> element elsewhere within the document.

Module header — *2. The TEI Header*

Attributes att.declarable (*@default*)

@datum supplies a commonly used code name for the datum employed.

Status Mandatory when applicable

Datatype `data.enumerated`

Suggested values include: **WGS84** (World Geodetic System) a pair of numbers to be interpreted as latitude followed by longitude according to the World Geodetic System. **[Default]**

MGRS (Military Grid Reference System) the values supplied are geospatial entity object codes, based on Universal Transverse Mercator coordinates

OSGB36 (ordnance survey great britain) the value supplied is to be interpreted as a British National Grid Reference.

ED50 (European Datum coordinate system) the value supplied is to be

interpreted as latitude followed by longitude according to the European Datum coordinate system.

Used by model.encodingPart

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address binaryObject cb choice corr date del distinct email emph expan foreign gap gloss graphic hi index lb measure measureGrp mentioned milestone name note num orig pb ptr ref reg rs sic soCalled term time title unclear

dictionaries: lang oRef oVar pRef pVar

figures: formula

gaiji: g

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident specDesc specList tag val

textcrit: app witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element geoDecl
{
   att.global.attributes,
   att.declarable.attributes,
   attribute datum { "WGS84" | "MGRS" | "OSGB36" | "ED50" | xsd:Name }?,
   macro.phraseSeq}
```

Example

```
<geoDecl datum="OSGB36"/>
```

<geogFeat> (geographical feature name) contains a common noun identifying some geographical feature contained within a geographic name, such as valley, mount, etc.

Module namesdates — *13. Names, Dates, People, and Places*

Attributes att.naming (*@key*, *@ref*, *@nymRef*) att.typed (*@type*, *@subtype*) att.datableatt.datable.w3c (*@period*, *@when*, *@notBefore*, *@notAfter*, *@from*, *@to*) att.datable.iso (*@when-iso*, *@notBefore-iso*, *@notAfter-iso*, *@from-iso*, *@to-iso*)

Used by model.offsetLike

May contain

gaiji: g

Declaration

```
element geogFeat
{
   att.global.attributes,
   att.naming.attributes,
   att.typed.attributes,
   att.datable.w3c.attributes,
   att.datable.iso.attributes,
   macro.xtext}
```

Example

```
<geogName> The <geogFeat>vale</geogFeat>
of White Horse</geogName>
```

<geogName> (geographical name) a name associated with some geographical feature such as Windrush Valley or Mount Sinai.

Module namesdates — *13. Names, Dates, People, and Places*

Attributes att.naming (*@key*, *@ref*, *@nymRef*)

@type provides more culture- linguistic- or application- specific information used to categorize this name component.

Status Mandatory when applicable

Datatype `data.enumerated`

Values one of a set of codes defined for the application.

Used by model.placeNamePart

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address binaryObject cb choice corr date del distinct email emph expan foreign gap gloss graphic hi index lb measure measureGrp mentioned milestone name note num orig pb ptr ref reg rs sic soCalled term time title unclear

dictionaries: lang oRef oVar pRef pVar

figures: formula

gaiji: g

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident specDesc specList tag val

textcrit: app witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element geogName
{
   att.global.attributes,
   att.naming.attributes,
   attribute type { data.enumerated }?,
   macro.phraseSeq}
```

Example

```
<geogName>
  <geogFeat>Mount</geogFeat>
  <name>Sinai</name>
</geogName>
```

<gi> (element name) contains the name (generic identifier) of an element.

Module tagdocs — *22. Documentation Elements*

Attributes In addition to global attributes

@scheme supplies the name of the scheme in which this name is defined.

Status Optional

Datatype `data.enumerated`

Sample values include: **TEI** (text encoding initiative) this element is part of the TEI scheme. **[Default]**

DBK (docbook) this element is part of the Docbook scheme.

XX (unknown) this element is part of an unknown scheme.

Used by model.phrase.xml

May contain Character data only

Declaration

```
element gi
{
   att.global.attributes,
   attribute scheme { data.enumerated }?,
   text
}
```

Example

```
<p>The
<gi>xhtml:li</gi> element is roughly analogous to the
<gi>item</gi> element, as is the <gi  scheme="DBK">listItem</gi> element.</p>
```

This example shows the use of both a namespace prefix and the schema attribute as alternative ways of indicating that the gi in question is not a TEI element name: in practice only one method should be adopted.

<gloss> identifies a phrase or word used to provide a gloss or definition for some other word or phrase.

Module core — *3. Elements Available in All TEI Documents*

Attributes att.declaring (*@decls*) att.translatable (*@version*) att.typed (*@type*, *@subtype*)

@target identifies the associated <term> element by an absolute or relative URI reference
Status Optional
Datatype `data.pointer`
Values should be a valid URI reference that resolves to a <term> element

@cRef (canonical reference) identifies the associated <term> element using a canonical reference from a scheme defined in a <refsDecl> element in the TEI header
Status Optional
Datatype `data.pointer`
Values the result of applying the algorithm for the resolution of canonical references (described in section *16.2.5. Canonical References*) should be a valid URI reference that resolves to a <term> element
Note The <refsDecl> to use may be indicated with the **decls** attribute.

Used by model.emphLike model.glossLike

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address binaryObject cb choice corr date del distinct email emph expan foreign gap gloss graphic hi index lb measure measureGrp mentioned milestone name note num orig pb ptr ref reg rs sic soCalled term time title unclear

dictionaries: lang oRef oVar pRef pVar

figures: formula

gaiji: g

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident specDesc specList tag val

textcrit: app witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element gloss
{
   att.global.attributes,
   att.declaring.attributes,
   att.translatable.attributes,
   att.typed.attributes,
   ( attribute target { data.pointer }? | attribute cRef { data.pointer }? ),
   macro.phraseSeq}
```

Example

```
We may define <term  xml:id="tdpv"  rend="sc">discoursal point of view</term>
as
<gloss  target="#tdpv">the relationship, expressed through discourse
structure, between the implied author or some other addresser,
and the fiction.</gloss>
```

Note The **target** and **cRef** attributes are mutually exclusive.

<glyph> (character glyph) provides descriptive information about a character glyph.

Module gaiji — *5. Representation of Non-standard Characters and Glyphs*

Attributes Global attributes only

Used by charDecl

May contain

core: binaryObject desc gloss graphic note

figures: formula

gaiji: charProp glyphName mapping

tagdocs: altIdent equiv

textcrit: witDetail

Declaration

```
element glyph
{
   att.global.attributes,
   (
      glyphName?,
      model.glossLike*,
      charProp*,
      mapping*,
      model.graphicLike*,
      model.noteLike*
   )
}
```

Example

```
<glyph  xml:id="rstroke">
  <glyphName>LATIN SMALL LETTER R WITH A FUNNY STROKE</glyphName>
  <charProp>
```

```
    <localName>entity</localName>
    <value>rstroke</value>
  </charProp>
  <graphic  url="glyph-rstroke.png"/>
</glyph>
```

<glyphName> (character glyph name) contains the name of a glyph, expressed following Unicode conventions for character names.

Module gaiji — *5. Representation of Non-standard Characters and Glyphs*

Attributes Global attributes only

Used by glyph

May contain Character data only

Declaration `element glyphName { att.global.attributes, text }`

Example

```
<glyphName>CIRCLED IDEOGRAPH 4EBA</glyphName>
```

Note For characters of non-ideographic scripts, a name following the conventions for Unicode names should be chosen. For ideographic scripts, an Ideographic Description Sequence (IDS) as described in Chapter 10.1 of the Unicode Standard is recommended where possible. Projects working in similar fields are recommended to coordinate and publish their list of <glyphName>s to facilitate data exchange.

<gram> (grammatical information) within an entry in a dictionary or a terminological data file, contains grammatical information relating to a term, word, or form.

Module dictionaries — *9. Dictionaries*

Attributes att.lexicographic (*@expand*, *@norm*, *@split*, *@value*, *@orig*, *@location*, *@mergedIn*, *@opt*)

@type classifies the grammatical information given according to some convenient typology — in the case of terminological information, preferably the dictionary of data element types specified in ISO WD 12 620.

Status Optional

Datatype `data.enumerated`

Sample values include: **pos** (part of speech) any of the word classes to which a word may be assigned in a given language, based on form, meaning, or a combination of features, e.g. noun, verb, adjective, etc.

gen (gender) formal classification by which nouns and pronouns, and often accompanying modifiers, are grouped and inflected, or changed in form, so as to control certain syntactic relationships

num (number) grammatical number, e.g. singular, plural, dual, …

animate animate or inanimate

proper proper noun or common noun

Note A much fuller list of values for the **type** attribute may be generated from the

dictionary of data element types under preparation as ISO TC 37/SC 3/WD 12 620, Computational Aids in Terminology. See ISO 12 620 for fuller details.

Used by model.morphLike

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address bibl biblStruct binaryObject cb choice cit corr date del desc distinct email emph expan foreign gap gloss graphic hi index label lb list listBibl measure measureGrp mentioned milestone name note num orig pb ptr q quote ref reg rs said sic soCalled stage term time title unclear

dictionaries: lang oRef oVar pRef pVar

drama: camera caption castList move sound tech view

figures: figure formula table

gaiji: g

header: biblFull

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material msDesc origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName listNym listOrg listPerson listPlace nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att classSpec code eg egXML elementSpec gi ident listRef macroSpec moduleRef moduleSpec specDesc specGrp specGrpRef specList tag val

textcrit: app listWit witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element gram
{
   att.global.attributes,
   att.lexicographic.attributes,
   attribute type { data.enumerated }?,
   macro.paraContent}
```

Example

```
<entry>
  <form>
    <orth>pamplemousse</orth>
  </form>
  <gramGrp>
    <gram  type="pos">noun</gram>
    <gram  type="gen">masculine</gram>
```

```
    </gramGrp>
</entry>
```

Note In terminological data, the <gram> element usually refers to the most recently specified <term> or <otherForm> element. In flat term entries, the **group** and **depend** attributes may be used to indicate exceptions to this general rule. In dictionaries, the element typically relates to the form or forms with which it is grouped in a <form> or other grouping element.

<gramGrp>

(grammatical information group) groups morpho-syntactic information about a lexical item, e.g. <pos>, <gen>, <number>, <case>, or <iType> (inflectional class).

Module dictionaries — *9. Dictionaries*

Attributes att.lexicographic (*@expand*, *@norm*, *@split*, *@value*, *@orig*, *@location*, *@mergedIn*, *@opt*)

Used by model.entryPart.top model.entryPart model.gramPart

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address bibl biblStruct binaryObject cb choice cit corr date del desc distinct email emph expan foreign gap gloss graphic hi index label lb list listBibl measure measureGrp mentioned milestone name note num orig pb ptr q quote ref reg rs said sic soCalled stage term time title unclear

dictionaries: case colloc gen gram gramGrp iType lang lbl mood number oRef oVar pRef pVar per pos subc tns usg

drama: camera caption castList move sound tech view

figures: figure formula table

gaiji: g

header: biblFull

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material msDesc origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName listNym listOrg listPerson listPlace nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att classSpec code eg egXML elementSpec gi ident listRef macroSpec moduleRef moduleSpec specDesc specGrp specGrpRef specList tag val

textcrit: app listWit witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element gramGrp
{
```

```
   att.global.attributes,
   att.lexicographic.attributes,
   (
      text
    | model.gLike    | model.phrase    | model.inter    | model.gramPart    | model.global   )*
}
```

Example

```
<entry>
  <form>
    <orth>luire</orth>
  </form>
  <gramGrp>
    <pos>verb</pos>
    <subc>intransitive</subc>
  </gramGrp>
</entry>
```

<graph> encodes a graph, which is a collection of nodes, and arcs which connect the nodes.

Module nets — *19. Graphs, Networks, and Trees*

Attributes In addition to global attributes

@type describes the type of graph.

Status Recommended

Datatype `data.enumerated`

Suggested values include: **undirected** undirected graph

directed directed graph

transitionNetwork a directed graph with distinguished initial and final nodes

transducer a transition network with up to two labels on each arc

Note If **type** is specified as **undirected**, then the distinction between the **to** and **from** attributes of the <arc> tag is neutralized. Also, the **adj** attribute, rather than the **adjFrom** and **adjTo** attributes, should be used to encode pointers to the ends of the arcs. If **type** is specified as **directed** (or any other value which implies directionality), then the **adjFrom** and **adjTo** attributes should be used, instead of the **adj** attribute.

@order states the order of the graph, i.e., the number of its nodes.

Status Optional

Datatype `data.count`

Values A positive integer.

@size states the size of the graph, i.e., the number of its arcs.

Status Optional

Datatype `data.count`

Values A non-negative integer.

Used by model.divPart

May contain

analysis: interp interpGrp span spanGrp

certainty: certainty respons

core: cb gap index label lb milestone note pb

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp timeline

nets: arc node

spoken: incident kinesic pause shift vocal writing

textcrit: witDetail

transcr: addSpan damageSpan delSpan fw space

Declaration

```
element graph
{
   att.global.attributes,
   attribute type
   {
      "undirected" | "directed" | "transitionNetwork" | "transducer" | xsd:Name
   }?,
   attribute order { data.count }?,
   attribute size { data.count }?,
   (
      ( label, model.global* )?,
      (
         ( ( node, model.global* )+, ( arc, model.global* )* )
       | ( ( arc, model.global* )+, ( node, model.global* )+ )
      )
   )
}
```

Example

```
<graph
   xml:id="cug1"
   type="undirected"
   order="5"
   size="4"
   rend="LABEL-PLACE bottom center NODE-FRAME none ARC solid line">
  <label>Airline Connections in Southwestern USA</label>
  <node  xml:id="lax"  degree="2">
    <label>LAX</label>
  </node>
  <node  xml:id="lvg"  degree="2">
    <label>LVG</label>
  </node>
  <node  xml:id="phx"  degree="3">
    <label>PHX</label>
  </node>
  <node  xml:id="tus"  degree="1">
    <label>TUS</label>
  </node>
  <node  xml:id="cib"  degree="0">
    <label>CIB</label>
  </node>
  <arc  from="#lax"  to="#lvg"/>
  <arc  from="#lax"  to="#phx"/>
  <arc  from="#lvg"  to="#phx"/>
```

```
  <arc  from="#phx"  to="#tus"/>
</graph>
```

Note One or more nodes and zero or more arcs in any order.

<graphic/> indicates the location of an inline graphic, illustration, or figure.

Module core — *3. Elements Available in All TEI Documents*

Attributes att.internetMedia (*@mimeType*) att.declaring (*@decls*)

@width The display width of the image

Status Mandatory when applicable

Datatype `data.outputMeasurement`

@height The display height of the image

Status Mandatory when applicable

Datatype `data.outputMeasurement`

@scale A scale factor to be applied to the image to make it the desired display size

Status Mandatory when applicable

Datatype `data.numeric`

@url (uniform resource locator) A URL which refers to the image itself.

Status Mandatory when applicable

Datatype `data.pointer`

Used by model.graphicLike model.titlepagePart

May contain Empty element

Declaration

```
element graphic
{
   att.global.attributes,
   att.internetMedia.attributes,
   att.declaring.attributes,
   attribute width { data.outputMeasurement }?,
   attribute height { data.outputMeasurement }?,
   attribute scale { data.numeric }?,
   attribute url { data.pointer }?,
   empty
}
```

Example

```
<figure>
  <graphic  url="fig1.png"/>
  <head>Figure One: The View from the Bridge</head>
  <figDesc>A Whistleresque view showing four
     or five sailing boats in the foreground, and a
     series of buoys strung out between them.</figDesc>
</figure>
```

Note The **mimeType** attribute should be used to supply the MIME media type of the image specified by the **url** attribute.

<group> contains the body of a composite text, grouping together a sequence of distinct texts (or groups of such texts) which are regarded as a unit for some purpose, for example the collected works of an author, a sequence of prose essays, etc.

Module textstructure — *4. Default Text Structure*

Attributes att.declaring (*@decls*)

Used by floatingText group text

May contain

analysis: interp interpGrp span spanGrp

certainty: certainty respons

core: cb gap head index lb meeting milestone note pb

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp timeline

spoken: incident kinesic pause shift vocal writing

textcrit: witDetail

textstructure: argument byline closer dateline docAuthor docDate epigraph group opener postscript salute signed text trailer

transcr: addSpan damageSpan delSpan fw space

Declaration

```
element group
{
   att.global.attributes,
   att.declaring.attributes,
   (
      ( model.divTop | model.global )*,
      ( ( text | group ), ( text | group | model.global )* ),
      model.divBottom*
   )
}
```

Example

```
<egXML><text>
<!-- Section on Alexander Pope starts -->
   <front>
<!-- biographical notice by editor -->
   </front>
   <group>
     <text>
<!-- first poem -->
     </text>
     <text>
<!-- second poem -->
     </text>
   </group>
</text>
<!-- end of Pope section-->
</egXML>
```

<handDesc> (description of hands) contains a description of all the different kinds of writing used in a manuscript.

Module msdescription — *10. Manuscript Description*

Attributes In addition to global attributes

@hands specifies the number of distinct hands identified within the manuscript

Status Optional

Datatype data.count

Values A whole number, or the string "many"

Used by model.physDescPart

May contain

core: p

header: handNote

linking: ab

msdescription: summary

Declaration

```
element handDesc
{
   att.global.attributes,
   attribute hands { data.count }?,
   ( model.pLike+ | ( summary?, handNote+ ) )
}
```

Example

```
<handDesc>
  <handNote scope="major">Written throughout in <term>angelicana formata</term>.</handNote>
</handDesc>
```

Example

```
<handDesc hands="2">
  <p>The manuscript is written in two contemporary hands, otherwise
     unknown, but clearly those of practised scribes. Hand I writes
     ff. 1r-22v and hand II ff. 23 and 24. Some scholars, notably
     Verner Dahlerup and Hreinn Benediktsson, have argued for a third hand
     on f. 24, but the evidence for this is insubstantial.</p>
</handDesc>
```

Example

```
<handDesc hands="2">
  <handNote xml:id="TSE" medium="typescript">Authorial typescript
  </handNote>
  <handNote xml:id="EP" medium="red-ink">Ezra Pound's
     annotations</handNote>
</handDesc>
```

<handNote> (note on hand) describes a particular style or hand distinguished within a manuscript.

Module header — *2. The TEI Header*

Attributes att.handFeatures (*@scribe*, *@script*, *@medium*, *@scope*)

Used by handDesc handNotes

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address bibl biblStruct binaryObject cb choice cit corr date del desc distinct email emph expan foreign gap gloss graphic hi index l label lb lg list listBibl measure measureGrp mentioned milestone name note num orig p pb ptr q quote ref reg rs said sic soCalled sp stage term time title unclear

dictionaries: lang oRef oVar pRef pVar

drama: camera caption castList move sound tech view

figures: figure formula table

gaiji: g

header: biblFull

iso-fs: fLib fs fvLib

linking: ab alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material msDesc origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName listNym listOrg listPerson listPlace nameLink offset orgName persName placeName region roleName settlement state surname

nets: eTree forest forestGrp graph tree

spoken: incident kinesic pause shift u vocal writing

tagdocs: att classSpec code eg egXML elementSpec gi ident listRef macroSpec moduleRef moduleSpec schemaSpec specDesc specGrp specGrpRef specList tag val

textcrit: app listWit witDetail

textstructure: floatingText

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element handNote
{
   att.global.attributes,
   att.handFeatures.attributes,
   macro.specialPara}
```

Example

```
<handNote  scope="sole">
  <p>Written in insular phase II half-uncial
     with interlinear Old English gloss in an Anglo-Saxon
     pointed minuscule.</p>
</handNote>
```

<handNotes> contains one or more <handNote> elements documenting the different hands identified within the source texts.

Module transcr — *11. Representation of Primary Sources*

Attributes Global attributes only

Used by model.profileDescPart

May contain

header: handNote

Declaration

```
element handNotes { att.global.attributes, handNote+ }
```

Example

```
<handNotes>
  <handNote xml:id="H1" script="copperplate" medium="brown-ink">Carefully written with regular
descenders</handNote>
  <handNote xml:id="H2" script="print" medium="pencil">Unschooled scrawl</handNote>
</handNotes>
```

<handShift/> marks the beginning of a sequence of text written in a new hand, or the beginning of a scribal stint.

Module transcr — *11. Representation of Primary Sources*

Attributes att.handFeatures (*@scribe*, *@script*, *@medium*, *@scope*)

@new identifies the new hand.

Status Recommended

Datatype `data.code`

Values must be one of the hand identifiers declared in the document header (see section *11.4.1. Document Hands*).

@resp signifies the editor or transcriber responsible for identifying the change of hand.

Status Recommended

Datatype `data.code`

Values must be one of the identifiers declared in the document header, associated with a person asserted as responsible for some aspect of the text's creation, transcription, editing, or encoding (see chapter *21. Certainty and Responsibility*).

Used by model.pPart.msdesc

May contain Empty element

Declaration

```
element handShift
{
   att.global.attributes,
   att.handFeatures.attributes,
   attribute new { data.code }?,
   attribute resp { data.code }?,
   empty
}
```

Example

```
<l>When wolde the cat dwelle in his ynne</l>
<handShift  medium="greenish-ink"/>
<l>And if the cattes skynne be slyk <handShift  medium="black-ink"/> and gaye</l>
```

Note The <handShift> element may be used either to denote a shift in the document hand (as from one scribe to another, on one writing style to another). Or, it may indicate a shift within a document hand, as a change of writing style, character or ink

---

<head> (heading) contains any type of heading, for example the title of a section, or the heading of a list, glossary, manuscript description, etc.

Module core — *3. Elements Available in All TEI Documents*

Attributes att.typed (*@type*, *@subtype*)

Used by model.headLike model.pLike.front

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address bibl biblStruct binaryObject cb choice cit corr date del desc distinct email emph expan foreign gap gloss graphic hi index label lb list listBibl measure measureGrp mentioned milestone name note num orig pb ptr q quote ref reg rs said sic soCalled stage term time title unclear

dictionaries: lang oRef oVar pRef pVar

drama: camera caption castList move sound tech view

figures: figure formula table

gaiji: g

header: biblFull

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material msDesc origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName listNym listOrg listPerson listPlace nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att classSpec code eg egXML elementSpec gi ident listRef macroSpec moduleRef moduleSpec specDesc specGrp specGrpRef specList tag val

textcrit: app listWit witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element head { att.global.attributes, att.typed.attributes, macro.paraContent }
```

Example The most common use for the <head> element is to mark the headings of sections. In older writings, the headings or incipits may be rather longer than usual in modern works. If a section has an explicit ending as well as a heading, it should be marked as a <trailer>, as in this example:

```
<div1  n="I"  type="book">
  <head>In the name of Christ here begins
     the first book of the ecclesiastical history of Georgius
     Florentinus, known as Gregory, Bishop of Tours.</head>
  <list>
    <head>Chapter-Headings</head>
  </list>
  <div2  type="section">
    <head>In the name of Christ here begins Book I of the
        history.</head>
    <p>Proposing as I do ...</p>
    <p>From the Passion of our Lord until the death of Saint Martin
        four hundred and twelve years passed.</p>
    <trailer>Here ends the first Book, which covers five thousand,
        five hundred and ninety-six years from the beginning of the
        world down to the death of Saint Martin.</trailer>
  </div2>
</div1>
```

Example The <head> element is also used to mark headings of other units, such as lists:

```
With a few exceptions, connectives are equally useful in
all kinds of discourse: description, narration, exposition,
argument.
<list  type="simple">
  <head>Connectives</head>
  <item>above</item>
  <item>accordingly</item>
  <item>across from</item>
  <item>adjacent to</item>
  <item>again</item>
  <item>
<!-- ... -->
  </item>
</list>
```

Note The <head> element is used for headings at all levels; software which treats (e.g.) chapter headings, section headings, and list titles differently must determine the proper processing of a <head> element based on its structural position. A <head> occurring as the first element of a list is the title of that list; one occurring as the first element of a <div1> is the title of that chapter or section.

<headItem> (heading for list items) contains the heading for the item or gloss column in a glossary list or similar structured list.

Module core — *3. Elements Available in All TEI Documents*

Attributes Global attributes only

Used by list

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address binaryObject cb choice corr date del distinct email emph expan foreign gap gloss graphic hi index lb measure measureGrp mentioned milestone name note num orig pb ptr ref reg rs sic soCalled term time title unclear

dictionaries: lang oRef oVar pRef pVar

figures: formula

gaiji: g

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident specDesc specList tag val

textcrit: app witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element headItem { att.global.attributes, macro.phraseSeq }
```

Example

```
The simple, straightforward statement of an idea is
preferable to the use of a worn-out expression.
<list type="gloss">
  <headLabel rend="small caps">TRITE</headLabel>
  <headItem rend="small caps">SIMPLE, STRAIGHTFORWARD</headItem>
  <label>bury the hatchet</label>
  <item>stop fighting, make peace</item>
  <label>at loose ends</label>
  <item>disorganized</item>
  <label>on speaking terms</label>
  <item>friendly</item>
  <label>fair and square</label>
  <item>completely honest</item>
  <label>at death's door</label>
  <item>near death</item>
</list>
```

Note The <headItem> element may appear only if each item in the list is preceded by a <label>.

<headLabel>

<headLabel> (heading for list labels) contains the heading for the label or term column in a glossary list or similar structured list.

Module core — *3. Elements Available in All TEI Documents*

Attributes Global attributes only

Used by list

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address binaryObject cb choice corr date del distinct email emph expan foreign gap gloss graphic hi index lb measure measureGrp mentioned milestone name note num orig pb ptr ref reg rs sic soCalled term time title unclear

dictionaries: lang oRef oVar pRef pVar

figures: formula

gaiji: g

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident specDesc specList tag val

textcrit: app witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element headLabel { att.global.attributes, macro.phraseSeq }
```

Example

```
The simple, straightforward statement of an idea is
preferable to the use of a worn-out expression.
<list type="gloss">
  <headLabel rend="small caps">TRITE</headLabel>
  <headItem rend="small caps">SIMPLE, STRAIGHTFORWARD</headItem>
  <label>bury the hatchet</label>
  <item>stop fighting, make peace</item>
  <label>at loose ends</label>
  <item>disorganized</item>
  <label>on speaking terms</label>
  <item>friendly</item>
  <label>fair and square</label>
  <item>completely honest</item>
  <label>at death's door</label>
  <item>near death</item>
</list>
```

Note The <headLabel> element may appear only if each item in the list is preceded by a <label>.

<height> contains a measurement measured along the axis parallel to the spine.

Module msdescription — *10. Manuscript Description*

Attributes att.dimensions (*@unit*, *@quantity*, *@precision*, *@scope*)

Used by dimensions model.measureLike

May contain

gaiji: g

Declaration

```
element height
{
   att.global.attributes,
   att.dimensions.attributes,
   macro.xtext}
```

Example

```
<height unit="in" quantity="7"/>
```

<heraldry> contains a heraldic formula or phrase, typically found as part of a blazon, coat of arms, etc.

Module msdescription — *10. Manuscript Description*

Attributes Global attributes only

Used by model.pPart.msdesc

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address binaryObject cb choice corr date del distinct email emph expan foreign gap gloss graphic hi index lb measure measureGrp mentioned milestone name note num orig pb ptr ref reg rs sic soCalled term time title unclear

dictionaries: lang oRef oVar pRef pVar

figures: formula

gaiji: g

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident specDesc specList tag val

textcrit: app witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element heraldry { att.global.attributes, macro.phraseSeq }
```

Example

```
<p>Ownership stamp (xvii cent.) on i recto with the arms
<heraldry>A bull passant within a bordure bezanty,
    in chief a crescent for difference</heraldry> [Cole],
crest, and the legend <q>Cole Deum</q>.</p>
```

<hi> (highlighted) marks a word or phrase as graphically distinct from the surrounding text, for reasons concerning which no claim is made.

Module core — *3. Elements Available in All TEI Documents*

Attributes Global attributes only

Used by model.hiLike

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address bibl biblStruct binaryObject cb choice cit corr date del desc distinct email emph expan foreign gap gloss graphic hi index label lb list listBibl measure measureGrp mentioned milestone name note num orig pb ptr q quote ref reg rs said sic soCalled stage term time title unclear

dictionaries: lang oRef oVar pRef pVar

drama: camera caption castList move sound tech view

figures: figure formula table

gaiji: g

header: biblFull

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material msDesc origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName listNym listOrg listPerson listPlace nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att classSpec code eg egXML elementSpec gi ident listRef macroSpec moduleRef moduleSpec specDesc specGrp specGrpRef specList tag val

textcrit: app listWit witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element hi { att.global.attributes, macro.paraContent }
```

Example

```
<hi  rend="gothic">And this Indenture further witnesseth</hi>
that the said <hi  rend="italic">Walter Shandy</hi>, merchant,
in consideration of the said intended marriage ...
```

Source: [187]

<history> groups elements describing the full history of a manuscript or manuscript part.

Module msdescription — *10. Manuscript Description*

Attributes Global attributes only

Used by msDesc msPart

May contain

core: p

linking: ab

msdescription: acquisition origin provenance summary

Declaration

```
element history
{
   att.global.attributes,
   ( model.pLike+ | ( summary?, origin?, provenance*, acquisition? ) )
}
```

Example

```
<history>
  <origin>
    <p>Written in Durham during the mid twelfth
        century.</p>
  </origin>
  <provenance>
    <p>Recorded in two medieval
        catalogues of the books belonging to Durham Priory, made in 1391 and
        1405.</p>
    <p>Given to W. Olleyf by William Ebchester, Prior (1446-56)
        and later belonged to Henry Dalton, Prior of Holy Island (Lindisfarne)
        according to inscriptions on ff. 4v and 5.</p>
  </provenance>
  <acquisition>
    <p>Presented to Trinity College in 1738 by
        Thomas Gale and his son Roger.</p>
  </acquisition>
</history>
```

<hom> (homograph) groups information relating to one homograph within an entry.

Module dictionaries — *9. Dictionaries*

Attributes att.lexicographic (*@expand*, *@norm*, *@split*, *@value*, *@orig*, *@location*, *@mergedIn*, *@opt*)

Used by entry model.entryPart

May contain

analysis: interp interpGrp span spanGrp

certainty: certainty respons

core: cb cit gap index lb milestone note pb

dictionaries: def dictScrap etym form gramGrp re sense usg xr

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp timeline

spoken: incident kinesic pause shift vocal writing

textcrit: witDetail

transcr: addSpan damageSpan delSpan fw space

Declaration

```
element hom
{
   att.global.attributes,
   att.lexicographic.attributes,
   ( sense | model.entryPart.top | model.global )*
}
```

Example

```
<entry>
  <form>
    <orth>bray</orth>
    <pron>breI</pron>
  </form>
  <hom>
    <gramGrp>
      <pos>n</pos>
    </gramGrp>
    <def>cry of an ass; sound of a trumpet.</def>
  </hom>
  <hom>
    <gramGrp>
      <pos>vt</pos>
      <subc>VP2A</subc>
    </gramGrp>
    <def>make a cry or sound of this kind.</def>
  </hom>
</entry>
```

<hyph> (hyphenation) contains a hyphenated form of a dictionary headword, or hyphenation information in some other form.

Module dictionaries — *9. Dictionaries*

Attributes att.lexicographic (*@expand*, *@norm*, *@split*, *@value*, *@orig*, *@location*, *@mergedIn*, *@opt*)

Used by model.entryPart model.formPart

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address bibl biblStruct binaryObject cb choice cit corr date del desc distinct email emph expan foreign gap gloss graphic hi index label lb list listBibl measure measureGrp mentioned milestone name note num orig pb ptr q quote ref reg rs said sic soCalled stage term time title unclear

dictionaries: lang oRef oVar pRef pVar

drama: camera caption castList move sound tech view

figures: figure formula table

gaiji: g

header: biblFull

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material msDesc origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName listNym listOrg listPerson listPlace nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att classSpec code eg egXML elementSpec gi ident listRef macroSpec moduleRef moduleSpec specDesc specGrp specGrpRef specList tag val

textcrit: app listWit witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element hyph
{
   att.global.attributes,
   att.lexicographic.attributes,
   macro.paraContent}
```

Example

```
<entry>
  <form>
    <orth>competitor</orth>
    <hyph>com|peti|tor</hyph>
    <pron>k@m"petit@(r)</pron>
  </form>
</entry>
```

<hyphenation> summarizes the way in which hyphenation in a source text has been treated in an encoded version of it.

Module header — *2. The TEI Header*

Attributes att.declarable (*@default*)

@eol (end-of-line) indicates whether or not end-of-line hyphenation has been retained in a text.

Status Optional

Legal values are: **all** all end-of-line hyphenation has been retained, even though the lineation of the original may not have been.

some end-of-line hyphenation has been retained in some cases. [**Default**]

hard all soft end-of-line hyphenation has been removed: any remaining end-od-line hyphenation should be retained.

none all end-of-line hyphenation has been removed: any remaining hyphenation occurred within the line.

Used by model.editorialDeclPart

May contain

core: p

linking: ab

Declaration

```
element hyphenation
{
   att.global.attributes,
   att.declarable.attributes,
   attribute eol { "all" | "some" | "hard" | "none" }?,
   model.pLike+
}
```

Example

```
<hyphenation eol="some">
  <p>End-of-line hyphenation silently removed where appropriate</p>
</hyphenation>
```

<iNode> (intermediate (or internal) node) represents an intermediate (or internal) node of a tree.

Module nets — *19. Graphs, Networks, and Trees*

Attributes In addition to global attributes

@value provides the value of an intermediate node, which is a feature structure or other analytic element.

Status Required when applicable

Datatype `data.pointer`

Values A valid identifier of a feature structure or other analytic element.

@children provides a list of identifiers of the elements which are the children of the intermediate node.

Status **Required**

Datatype 1–∞ occurrences of `data.pointer` separated by whitespace

Values A list of identifiers.

@parent provides the identifier of the element which is the parent of this node.

Status Optional

Datatype `data.pointer`

Values The identifier of the parent node.

@ord (ordered) indicates whether or not the internal node is ordered.

Status Optional

Datatype `data.xTruthValue`

Note The value **true** indicates that the children of the intermediate node are ordered, whereas **false** indicates the are unordered.Use if and only if **ord** is specified as **partial** on the <tree> element and the intermediate node has more than one child.

@follow provides an identifier of the element which this node follows.

Status Required when applicable

Datatype `data.pointer`

Values The identifier of another intermediate node or leaf of the tree.

Note If the tree is unordered or partially ordered, this attribute has the property of fixing the relative order of the intermediate node and the element which is the value of the attribute.

@outDegree gives the out degree of an intermediate node, the number of its children.

Status Optional

Datatype `data.count`

Values A nonnegative integer.

Note The in degree of an intermediate node is always 1.

Used by tree

May contain

core: label

Declaration

```
element iNode
{
   att.global.attributes,
   attribute value { data.pointer }?,
   attribute children { list { data.pointer+ } },
   attribute parent { data.pointer }?,
   attribute ord { data.xTruthValue }?,
   attribute follow { data.pointer }?,
   attribute outDegree { data.count }?,
   label?
}
```

Example

```
<iNode
   xml:id="pt1"
   children="#GD-UP1"
   parent="#GD-VB1"
   follow="#GD-PN1"
   outDegree="1">
  <label>PT</label>
</iNode>
```

<iType> (inflectional class) indicates the inflectional class associated with a lexical item.

Module dictionaries — *9. Dictionaries*

Attributes att.lexicographic (*@expand*, *@norm*, *@split*, *@value*, *@orig*, *@location*, *@mergedIn*, *@opt*)

@type indicates the type of indicator used to specify the inflection class, when it is necessary to distinguish between the usual abbreviated indications (e.g. *inv*) and other kinds of indicators, such as special codes referring to conjugation patterns, etc.

Status Optional

Datatype `data.enumerated`

Sample values include: **abbrev** abbreviated indicator

verbTable coded reference to a table of verbs

Note This element is synonymous with <**gram type='inflectional type'**>.

Used by model.entryPart model.morphLike

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address bibl biblStruct binaryObject cb choice cit corr date del desc distinct email emph expan foreign gap gloss graphic hi index label lb list listBibl measure measureGrp mentioned milestone name note num orig pb ptr q quote ref reg rs said sic soCalled stage term time title unclear

dictionaries: lang oRef oVar pRef pVar

drama: camera caption castList move sound tech view

figures: figure formula table

gaiji: g

header: biblFull

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material msDesc origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName listNym listOrg listPerson listPlace nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att classSpec code eg egXML elementSpec gi ident listRef macroSpec moduleRef moduleSpec specDesc specGrp specGrpRef specList tag val

textcrit: app listWit witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element iType
{
   att.global.attributes,
   att.lexicographic.attributes,
```

```
   attribute type { data.enumerated }?,
   macro.paraContent}
```

Example

```
<form>
  <orth>horrifier</orth>
  <pron>ORifje</pron>
  <iType  type="vbtable">7</iType>
</form>
```

Note May contain character data and phrase-level elements. Typical content will be *invariant*, *n 3* etc.

<ident> (identifier) contains an identifier or name for an object of some kind in a formal language.

Module tagdocs — *22. Documentation Elements*

Attributes att.typed (*@type*, *@subtype*)

Used by model.emphLike

May contain Character data only

Declaration

```
element ident { att.global.attributes, att.typed.attributes, text }
```

Example

```
<ident  type="namespace">http://www.tei-c.org/ns/Examples</ident>
```

Note In running prose, this element may be used for any kind of identifier in any formal language.

<idno> (identifying number) supplies any standard or non-standard number used to identify a bibliographic item.

Module header — *2. The TEI Header*

Attributes In addition to global attributes

@type categorizes the number, for example as an ISBN or other standard series.

Status Optional

Datatype `data.enumerated`

Values A name or abbreviation indicating what type of identifying number is given (e.g. ISBN, LCCN).

Used by altIdentifier biblStruct msIdentifier seriesStmt model.biblPart model.publicationStmtPart

May contain Character data only

Declaration

```
element idno
{
   att.global.attributes,
   attribute type { data.enumerated }?,
   text
}
```

Example

```
<idno type="ISSN">0143-3385</idno>
<idno type="OTA">116</idno>
```

<if> defines a conditional default value for a feature; the condition is specified as a feature structure, and is met if it subsumes the feature structure in the text for which a default value is sought.

Module iso-fs — *18. Feature Structures*

Attributes Global attributes only

Used by vDefault

May contain

iso-fs: binary default f fs numeric string symbol then vAlt vColl vLabel vMerge vNot

Declaration

```
element if
{
   att.global.attributes,
   ( ( fs | f ), then, ( model.featureVal ) )
}
```

Example

```
<vDefault>
  <if>
    <fs>
      <f name="VFORM">
        <symbol value="INF"/>
      </f>
      <f name="SUBJ">
        <binary value="true"/>
      </f>
    </fs>
    <then/>
    <symbol value="for"/>
  </if>
</vDefault>
```

Note May contain a feature structure, followed by a feature value; the two are separated by a <then> element.

<iff/> (if and only if) separates the condition from the consequence in a bicond element.

Module iso-fs — *18. Feature Structures*

Attributes Global attributes only

Used by bicond

May contain Empty element

Declaration

```
element iff { att.global.attributes, empty }
```

Example

```
<bicond>
  <fs>
    <f  name="FOO">
      <symbol  value="42"/>
    </f>
  </fs>
  <iff/>
  <fs>
    <f  name="BAR">
      <binary  value="true"/>
    </f>
  </fs>
</bicond>
```

Note This element is provided primarily to enhance the human readability of the feature-system declaration.

<imprimatur> contains a formal statement authorizing the publication of a work, sometimes required to appear on a title page or its verso.

Module textstructure — *4. Default Text Structure*

Attributes Global attributes only

Used by model.titlepagePart

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address bibl biblStruct binaryObject cb choice cit corr date del desc distinct email emph expan foreign gap gloss graphic hi index label lb list listBibl measure measureGrp mentioned milestone name note num orig pb ptr q quote ref reg rs said sic soCalled stage term time title unclear

dictionaries: lang oRef oVar pRef pVar

drama: camera caption castList move sound tech view

figures: figure formula table

gaiji: g

header: biblFull

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material msDesc origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName listNym listOrg listPerson listPlace nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att classSpec code eg egXML elementSpec gi ident listRef macroSpec moduleRef moduleSpec specDesc specGrp specGrpRef specList tag val

textcrit: app listWit witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element imprimatur { att.global.attributes, macro.paraContent }
```

Example

```
<imprimatur>Licensed and entred acording to Order.</imprimatur>
```

<imprint> groups information relating to the publication or distribution of a bibliographic item.

Module core — *3. Elements Available in All TEI Documents*

Attributes Global attributes only

Used by monogr

May contain

analysis: interp interpGrp span spanGrp

certainty: certainty respons

core: biblScope cb date gap index lb milestone note pb pubPlace publisher time

header: distributor

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp timeline

spoken: incident kinesic pause shift vocal writing

textcrit: witDetail

transcr: addSpan damageSpan delSpan fw space

Declaration

```
element imprint
{
   att.global.attributes,
   ( ( ( model.imprintPart ) | ( model.dateLike ) ), model.global* )+
}
```

Example

```
<imprint>
  <pubPlace>Oxford</pubPlace>
  <publisher>Clarendon Press</publisher>
  <date>1987</date>
</imprint>
```

<incident> any phenomenon or occurrence, not necessarily vocalized or communicative, for example incidental noises or other events affecting communication.

Module spoken — *8. Transcriptions of Speech*

Attributes att.timed (*@start*, *@end*) att.duration.w3c (*@dur*) att.typed (*@type*, *@subtype*) att.ascribed (*@who*)

Used by model.global.spoken

May contain

core: desc gloss

tagdocs: altIdent equiv

Declaration

```
element incident
{
   att.global.attributes,
   att.timed.attributes,
   att.duration.w3c.attributes,
   att.typed.attributes,
   att.ascribed.attributes,
   model.glossLike*
}
```

Example

```
<incident>
  <desc>ceiling collapses</desc>
</incident>
```

<incipit> contains the incipit of a manuscript item, that is the opening words of the text proper, exclusive of any rubric which might precede it, of sufficient length to identify the work uniquely; such incipts were, in fomer times, frequently used a means of reference to a work, in place of a title.

Module msdescription — *10. Manuscript Description*

Attributes att.typed (*@type*, *@subtype*) att.msExcerpt (*@defective*)

Used by msItemStruct model.msItemPart

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address binaryObject cb choice corr date del distinct email emph expan foreign gap gloss graphic hi index lb measure measureGrp mentioned milestone name note num orig pb ptr ref reg rs sic soCalled term time title unclear

dictionaries: lang oRef oVar pRef pVar

figures: formula

gaiji: g

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident specDesc specList tag val

textcrit: app witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element incipit
{
   att.global.attributes,
   att.typed.attributes,
   att.msExcerpt.attributes,
   macro.phraseSeq}
```

Example

```
<incipit>Pater noster qui es in celis</incipit>
<incipit  defective="true">tatem dedit hominibus alleluia.</incipit>
<incipit  type="biblical">Ghif ons huden onse dagelix broet</incipit>
<incipit>O ongehoerde gewerdighe christi</incipit>
<incipit  type="lemma">Firmiter</incipit>
<incipit>Ideo dicit firmiter quia
ordo fidei nostre probari non potest</incipit>
```

<index> (index entry) marks a location to be indexed for whatever purpose.

Module core — *3. Elements Available in All TEI Documents*

Attributes att.spanning (*@spanTo*)

@indexName supplies a name to specify which index (of several) the index entry belongs to.

Status Optional

Datatype `data.name`

Values an application-specific name, consisting of Unicode characters only.

Note This attribute makes it possible to create multiple indexes for a text.

Used by index model.global.meta

May contain

core: index term

Declaration

```
element index
{
   att.global.attributes,
   att.spanning.attributes,
   attribute indexName { data.name }?,
   ( term, index? )*
}
```

Example

```
David's other principal backer, Josiah ha-Kohen
<index  indexName="NAMES">
  <term>Josiah ha-Kohen b. Azarya</term>
</index> b. Azarya, son of one of the last gaons of Sura
<index  indexName="PLACES">
  <term>Sura</term>
</index> was David's own first cousin.
```

<institution> contains the name of an organization such as a university or library, with which a manuscript is identified, generally its holding institution.

Module msdescription — *10. Manuscript Description*

Attributes att.naming (*@key*, *@ref*, *@nymRef*)

Used by altIdentifier msIdentifier

May contain

gaiji: g

Declaration

```
element institution
{
   att.global.attributes,
   att.naming.attributes,
   macro.xtext}
```

Example

```
<msIdentifier>
  <settlement>Oxford</settlement>
  <institution>University of Oxford</institution>
  <repository>Bodleian Library</repository>
  <idno>MS. Bodley 406</idno>
</msIdentifier>
```

<interaction> describes the extent, cardinality and nature of any interaction among those producing and experiencing the text, for example in the form of response or interjection, commentary, etc.

Module corpus — *15. Language Corpora*

Attributes In addition to global attributes

@type specifies the degree of interaction between active and passive participants in the text.

Status Optional

Legal values are: **none** no interaction of any kind, e.g. a monologue

partial some degree of interaction, e.g. a monologue with set responses

complete complete interaction, e.g. a face to face conversation

inapplicable this parameter is inappropriate or inapplicable in this case

@active specifies the number of active participants (or addressors) producing parts of the text.

Status Optional

Datatype `data.enumerated`

Suggested values include: **singular** a single addressor

plural many addressors

corporate a corporate addressor

unknown number of addressors unknown or unspecifiable

@passive specifies the number of passive participants (or addressees) to whom a text is directed or in whose presence it is created or performed.

Status Optional

Datatype `data.enumerated`

Suggested values include: **self** text is addressed to the originator e.g. a diary

single text is addressed to one other person e.g. a personal letter

many text is addressed to a countable number of others e.g. a conversation in which all participants are identified

group text is addressed to an undefined but fixed number of participants e.g. a lecture

world text is addressed to an undefined and indeterminately large number e.g. a published book

Used by model.textDescPart

May contain

analysis: interp interpGrp span spanGrp

certainty: certainty respons

core: abbr address cb choice date distinct email emph expan foreign gap gloss index lb measure measureGrp mentioned milestone name note num pb ptr ref rs soCalled term time title

dictionaries: lang

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident tag val

textcrit: witDetail

transcr: addSpan am damageSpan delSpan ex fw handShift space subst

Declaration

```
element interaction
{
   att.global.attributes,
   attribute type { "none" | "partial" | "complete" | "inapplicable" }?,
   attribute active
   {
      "singular" | "plural" | "corporate" | "unknown" | xsd:Name
   }?,
```

```
   attribute passive
   {
      "self" | "single" | "many" | "group" | "world" | xsd:Name
   }?,
   macro.phraseSeq.limited}
```

Example

```
<interaction type="complete" active="plural" passive="many"/>
```

Example

```
<interaction type="none" active="singular" passive="group"/>
```

<interp> (interpretation) summarizes a specific interpretative annotation which can be linked to a span of text.

Module analysis — *17. Simple Analytic Mechanisms*

Attributes att.interpLike (*@resp*, *@type*, *@inst*)

Used by interpGrp model.global.meta

May contain

core: desc gloss

gaiji: g

tagdocs: altIdent equiv

Declaration

```
element interp
{
   att.global.attributes,
   att.interpLike.attributes,
   ( text | model.gLike | model.glossLike )*
}
```

Example

```
<interp type="structuralunit">aftermath</interp>
```

<interpGrp> (interpretation group) collects together a set of related interpretations which share responsibility or type.

Module analysis — *17. Simple Analytic Mechanisms*

Attributes att.interpLike (*@resp*, *@type*, *@inst*)

Used by model.global.meta

May contain

analysis: interp

core: desc gloss

tagdocs: altIdent equiv

Declaration

```
element interpGrp
{
   att.global.attributes,
   att.interpLike.attributes,
   ( model.glossLike*, interp+ )
}
```

Example

```
<interpGrp  resp="#TMA"  type="structuralunit">
  <desc>basic structural organization</desc>
  <interp  xml:id="I1">introduction</interp>
  <interp  xml:id="I2">conflict</interp>
  <interp  xml:id="I3">climax</interp>
  <interp  xml:id="I4">revenge</interp>
  <interp  xml:id="I5">reconciliation</interp>
  <interp  xml:id="I6">aftermath</interp>
</interpGrp>
<bibl  xml:id="TMA">
<!-- bibliographic citation for source of this interpretive framework -->
</bibl>
```

Note Any number of <interp> elements.

<interpretation> describes the scope of any analytic or interpretive information added to the text in addition to the transcription.

Module header — *2. The TEI Header*

Attributes att.declarable (*@default*)

Used by model.editorialDeclPart

May contain

core: p

linking: ab

Declaration

```
element interpretation
{
   att.global.attributes,
   att.declarable.attributes,
   model.pLike+
}
```

Example

```
<interpretation>
  <p>The part of speech analysis applied throughout section 4 was
     added by hand and has not been checked</p>
</interpretation>
```

<item> contains one component of a list.

Module core — *3. Elements Available in All TEI Documents*

Attributes Global attributes only

Used by list

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address bibl biblStruct binaryObject cb choice cit corr date del desc distinct email emph expan foreign gap gloss graphic hi index l label lb lg list listBibl measure measureGrp mentioned milestone name note num orig p pb ptr q quote ref reg rs said sic soCalled sp stage term time title unclear

dictionaries: lang oRef oVar pRef pVar

drama: camera caption castList move sound tech view

figures: figure formula table

gaiji: g

header: biblFull

iso-fs: fLib fs fvLib

linking: ab alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material msDesc origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName listNym listOrg listPerson listPlace nameLink offset orgName persName placeName region roleName settlement state surname

nets: eTree forest forestGrp graph tree

spoken: incident kinesic pause shift u vocal writing

tagdocs: att classSpec code eg egXML elementSpec gi ident listRef macroSpec moduleRef moduleSpec schemaSpec specDesc specGrp specGrpRef specList tag val

textcrit: app listWit witDetail

textstructure: floatingText

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element item { att.global.attributes, macro.specialPara }
```

Example

```
<list type="ordered">
  <head>Here begin the chapter headings of Book IV</head>
  <item n="4.1">The death of Queen Clotild.</item>
  <item n="4.2">How King Lothar wanted to appropriate
     one third of the Church revenues.</item>
  <item n="4.3">The wives and children of Lothar.</item>
  <item n="4.4">The Counts of the Bretons.</item>
  <item n="4.5">Saint Gall the Bishop.</item>
  <item n="4.6">The priest Cato.</item>
```

```
  <item> ...</item>
</list>
```

Note May contain simple prose or a sequence of chunks.Whatever string of characters is used to label a list item in the copy text may be used as the value of the global **n** attribute, but it is not required that numbering be recorded explicitly. In ordered lists, the **n** attribute on the <item> element is by definition synonymous with the use of the <label> element to record the enumerator of the list item. In glossary lists, however, the term being defined should be given with the <label> element, not **n**.

<join> identifies a possibly fragmented segment of text, by pointing at the possibly discontiguous elements which compose it.

Module linking — *16. Linking, Segmentation, and Alignment*

Attributes att.pointing (*@type*, *@evaluate*)

@targets specifies the identifiers of the elements or passages to be joined into a virtual element.

Status **Required**

Datatype 2–∞ occurrences of `data.pointer` separated by whitespace

Values two or more pointers (URIs), separated by whitespace

@result specifies the name of an element which this aggregation may be understood to represent.

Status Optional

Datatype `data.name`

Values The generic identifier of an element in the current DTD.

@scope indicates whether the targets to be joined include the entire element indicated (the entire subtree including its root), or just the children of the target (the branches of the subtree).

Status Recommended

Legal values are: **root** the rooted subtrees indicated by the targets attribute are joined, each subtree become a child of the virtual element created by the join **[Default]**

branches the children of the subtrees indicated by the targets attribute become the children of the virtual element (i.e. the roots of the subtrees are discarded)

Used by joinGrp model.global.meta

May contain

core: desc gloss

tagdocs: altIdent equiv

Declaration

```
element join
{
   att.global.attributes,
   att.pointing.attributes,
   attribute targets { list { data.pointer, data.pointer+ } },
```

```
   attribute result { data.name }?,
   attribute scope { "root" | "branches" }?,
   model.glossLike*
}
```

Example The following example is discussed in section *16.7. Aggregation*:

```
<sp>
  <speaker>Hughie</speaker>
  <p>How does it go?
  <q>
     <l  xml:id="frog_x1">da-da-da</l>
     <l  xml:id="frog_l2">gets a new frog</l>
     <l>...</l>
    </q>
  </p>
</sp>
<sp>
  <speaker>Louie</speaker>
  <p>
    <q>
     <l  xml:id="frog_l1">When the old pond</l>
     <l>...</l>
    </q>
  </p>
</sp>
<sp>
  <speaker>Dewey</speaker>
  <p>
    <q>...
    <l  xml:id="frog_l3">It's a new pond.</l>
    </q>
  </p>
  <join  targets="#frog_l1 #frog_l2 #frog_l3"  result="lg"  scope="root"/>
</sp>
```

The <join> element here identifies a linegroup (<lg>) comprising the three lines indicated by the **targets** attribute. The value `root` for the **scope** attribute indicates that the resulting virtual element contains the three <l> elements linked to at #frog_l1 #frog_l2 #frog_l3, rather than their character data content.

Example In this example, the attribute **scope** is specified with the value of `branches` to indicate that the virtual list being constructed is to be made by taking the lists indicated by the **targets** attribute of the <join> element, discarding the <list> tags which enclose them, and combining the items contained within the lists into a single virtual list:

```
<p>Southern dialect (my own variety, at least) has only
<list  xml:id="LP1">
   <item>
     <s>I done gone</s>
   </item>
   <item>
     <s>I done went</s>
   </item>
  </list> whereas Negro Non-Standard basilect has both these and
<list  xml:id="LP2">
   <item>
```

```
      <s>I done go</s>
    </item>
  </list>.</p>
<p>White Southern dialect also has
<list  xml:id="LP3">
    <item>
      <s>I've done gone</s>
    </item>
    <item>
      <s>I've done went</s>
    </item>
  </list> which, when they occur in Negro dialect, should probably
be considered as borrowings from other varieties of
English.</p>
<join
    result="list"
    xml:id="LST1"
    targets="#LP1 #LP2 #LP3"
    scope="branches">
  <desc>Sample sentences in Southern speech</desc>
</join>
```

<joinGrp> (join group) groups a collection of join elements and possibly pointers.

Module linking — *16. Linking, Segmentation, and Alignment*

Attributes att.pointing.group (*@domains*, *@targFunc*) att.pointing (*@type*, *@evaluate*)

@result describes the result of the joins gathered in this collection.

Status Optional

Datatype `data.name`

Values supplies the default value for the **result** on each <join> included within the group.

Used by model.global.meta

May contain

core: desc gloss ptr

linking: join

tagdocs: altIdent equiv

Declaration

```
element joinGrp
{
   att.global.attributes,
   att.pointing.group.attributes,
   att.pointing.attributes,
   attribute result { data.name }?,
   ( model.glossLike*, ( join | ptr )+ )
}
```

Example

```
<joinGrp domains="zuitxt zuitxt zuitxt" result="q">
  <join targets="#zuiq1 #zuiq2 #zuiq6"/>
  <join targets="#zuiq3 #zuiq4 #zuiq5"/>
</joinGrp>
```

Note Any number of <join> or <ptr> elements.

<keywords> contains a list of keywords or phrases identifying the topic or nature of a text.

Module header — *2. The TEI Header*

Attributes In addition to global attributes

@scheme identifies the controlled vocabulary within which the set of keywords concerned is defined.

Status **Required**

Datatype `data.pointer`

Values Usually this will indicate an external website or other location where the scheme is documented.

Used by textClass

May contain

core: list term

Declaration

```
element keywords
{
   att.global.attributes,
   attribute scheme { data.pointer },
   ( term+ | list )
}
```

Example

```
<keywords scheme="http://classificationweb.net">
  <list>
    <item>Babbage, Charles</item>
    <item>Mathematicians - Great Britain - Biography</item>
  </list>
</keywords>
```

<kinesic> any communicative phenomenon, not necessarily vocalized, for example a gesture, frown, etc.

Module spoken — *8. Transcriptions of Speech*

Attributes att.timed (*@start*, *@end*) att.duration.w3c (*@dur*) att.typed (*@type*, *@subtype*) att.ascribed (*@who*)

@iterated indicates whether or not the phenomenon is repeated.

Status Optional

Datatype `data.xTruthValue`

Note The value **true** indicates that the kinesic is repeated several times rather than occurring only once.

Used by model.global.spoken

May contain

core: desc gloss

tagdocs: altIdent equiv

Declaration

```
element kinesic
{
   att.global.attributes,
   att.timed.attributes,
   att.duration.w3c.attributes,
   att.typed.attributes,
   att.ascribed.attributes,
   attribute iterated { data.xTruthValue }?,
   model.glossLike*
}
```

Example

```
<kinesic dur="PT1.5S" iterated="true" type="reinforcing">
  <desc>nodding head vigorously</desc>
</kinesic>
```

<l> (verse line) contains a single, possibly incomplete, line of verse.

Module core — *3. Elements Available in All TEI Documents*

Attributes att.metrical (*@met*, *@real*, *@rhyme*) att.enjamb (*@enjamb*)

@part specifies whether or not the line is metrically complete.

Status Mandatory when applicable

Legal values are: **Y** (yes) the line is metrically incomplete

N (no) either the line is complete, or no claim is made as to its completeness **[Default]**

I (initial) the initial part of an incomplete line

M (medial) a medial part of an incomplete line

F (final) the final part of an incomplete line

Note The values **I**, **M**, or **F** should be used only where it is clear how the line is to be reconstituted.

Used by model.lLike

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address bibl biblStruct binaryObject cb choice cit corr date del desc distinct email emph expan foreign gap gloss graphic hi index label lb list listBibl measure measureGrp mentioned milestone name note num orig pb ptr q quote ref reg rs said sic soCalled stage term time title unclear

dictionaries: lang oRef oVar pRef pVar

drama: camera caption castList move sound tech view

figures: figure formula table

gaiji: g

header: biblFull

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material msDesc origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName listNym listOrg listPerson listPlace nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att classSpec code eg egXML elementSpec gi ident listRef macroSpec moduleRef moduleSpec specDesc specGrp specGrpRef specList tag val

textcrit: app listWit witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element l
{
   att.global.attributes,
   att.metrical.attributes,
   att.enjamb.attributes,
   attribute part { "Y" | "N" | "I" | "M" | "F" }?,
   macro.paraContent}
```

Example

```
<l met="-/-/-/-/-/" part="Y"/>
```

<label> contains the label associated with an item in a list; in glossaries, marks the term being defined.

Module core — *3. Elements Available in All TEI Documents*

Attributes Global attributes only

Used by arc eLeaf eTree graph iNode leaf list node root tree triangle model.labelLike

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address binaryObject cb choice corr date del distinct email emph expan foreign gap gloss graphic hi index lb measure measureGrp mentioned milestone name note num orig pb ptr ref reg rs sic soCalled term time title unclear

dictionaries: lang oRef oVar pRef pVar

figures: formula

gaiji: g

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident specDesc specList tag val

textcrit: app witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element label { att.global.attributes, macro.phraseSeq }
```

Example Labels are most commonly used for the headwords in glossary lists; note the use of the global **xml:lang** attribute to set the default language of the glossary list to Middle English, and identify the glosses and headings as modern English or Latin:

```
<list type="gloss" xml:lang="enm">
  <head xml:lang="en">Vocabulary</head>
  <headLabel xml:lang="en">Middle English</headLabel>
  <headItem xml:lang="en">New English</headItem>
  <label>nu</label>
  <item xml:lang="en">now</item>
  <label>lhude</label>
  <item xml:lang="en">loudly</item>
  <label>bloweth</label>
  <item xml:lang="en">blooms</item>
  <label>med</label>
  <item xml:lang="en">meadow</item>
  <label>wude</label>
  <item xml:lang="en">wood</item>
  <label>awe</label>
  <item xml:lang="en">ewe</item>
  <label>lhouth</label>
  <item xml:lang="en">lows</item>
  <label>sterteth</label>
  <item xml:lang="en">bounds, frisks
     (cf. <cit>
      <ref>Chaucer, K.T.644</ref>
      <quote>a courser,
      <term>sterting</term>as the fyr</quote>
    </cit>
  </item>
  <label>verteth</label>
  <item xml:lang="la">pedit</item>
  <label>murie</label>
  <item xml:lang="en">merrily</item>
  <label>swik</label>
  <item xml:lang="en">cease</item>
  <label>naver</label>
```

```
  <item  xml:lang="en">never</item>
</list>
```

Example Labels may also be used to record explicitly the numbers or letters which mark list items in ordered lists, as in this extract from Gibbon's *Autobiography*. In this usage the <label> element is synonymous with the **n** attribute on the <item> element:

```
I will add two facts, which have seldom occurred in
the composition of six, or at least of five quartos.
<list  rend="runon"  type="ordered">
  <label>(1)</label>
  <item>My first rough manuscript, without any
     intermediate copy, has been sent to the press.</item>
  <label>(2) </label>
  <item>Not a sheet has been seen by any human
     eyes, excepting those of the author and the printer:
     the faults and the merits are exclusively my own.</item>
</list>
```

Example Labels may also be used for other structured list items, as in this extract from the journal of Edward Gibbon:

```
<list  type="gloss">
  <label>March 1757.</label>
  <item>I wrote some critical observations upon Plautus.</item>
  <label>March 8th.</label>
  <item>I wrote a long dissertation upon some lines of Virgil.</item>
  <label>June.</label>
  <item>I saw Mademoiselle Curchod —
  <q  xml:lang="la">Omnia vincit amor, et nos cedamus amori.</q>
  </item>
  <label>August.</label>
  <item>I went to Crassy, and staid two days.</item>
</list>
```

<lacunaEnd/> indicates the end of a lacuna in a mostly complete textual witness.

Module textcrit — *12. Critical Apparatus*

Attributes att.rdgPart (*@wit*)

Used by model.rdgPart

May contain Empty element

Declaration

```
element lacunaEnd { att.global.attributes, att.rdgPart.attributes, empty }
```

Example

```
<rdg  wit="#X">
  <lacunaEnd/>auctorite
</rdg>
```

<lacunaStart/> indicates the beginning of a lacuna in the text of a mostly complete textual witness.

Module textcrit — *12. Critical Apparatus*

Attributes att.rdgPart (*@wit*)

Used by model.rdgPart

May contain Empty element

Declaration

```
element lacunaStart { att.global.attributes, att.rdgPart.attributes, empty }
```

Example

```
<app>
  <lem  wit="#El #Hg">Experience</lem>
  <rdg  wit="#Ha4">Ex<g  ref="#per"/>
    <lacunaStart/>
  </rdg>
</app>
```

<lang> (language name) name of a language mentioned in etymological or other linguistic discussion.

Module dictionaries — *9. Dictionaries*

Attributes att.lexicographic (*@expand*, *@norm*, *@split*, *@value*, *@orig*, *@location*, *@mergedIn*, *@opt*)

Used by model.nameLike

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address bibl biblStruct binaryObject cb choice cit corr date del desc distinct email emph expan foreign gap gloss graphic hi index label lb list listBibl measure measureGrp mentioned milestone name note num orig pb ptr q quote ref reg rs said sic soCalled stage term time title unclear

dictionaries: lang oRef oVar pRef pVar

drama: camera caption castList move sound tech view

figures: figure formula table

gaiji: g

header: biblFull

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material msDesc origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName listNym listOrg listPerson listPlace nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att classSpec code eg egXML elementSpec gi ident listRef macroSpec moduleRef moduleSpec specDesc specGrp specGrpRef specList tag val

textcrit: app listWit witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element lang
{
   att.global.attributes,
   att.lexicographic.attributes,
   macro.paraContent}
```

Example

```
<entry>
  <form>
    <orth>publish</orth> ... </form>
  <etym>
    <lang>ME.</lang>
    <mentioned>publisshen</mentioned>, <lang>F.</lang>
    <mentioned>publier</mentioned>, <lang>L.</lang>
    <mentioned>publicare, publicatum</mentioned>. <xr>See <ref>public</ref>; cf.
    <ref>2d -ish</ref>.</xr>
  </etym>
</entry>
```

Note May contain character data mixed with phrase-level elements.

<langKnowledge>

<langKnowledge> (language knowledge) summarizes the state of a person's linguistic knowledge, either as prose or by a list of <langKnown> elements.

Module namesdates — *13. Names, Dates, People, and Places*

Attributes att.datableatt.datable.w3c (*@period*, *@when*, *@notBefore*, *@notAfter*, *@from*, *@to*) att.datable.iso (*@when-iso*, *@notBefore-iso*, *@notAfter-iso*, *@from-iso*, *@to-iso*) att.editLike (*@cert*, *@resp*, *@evidence*, *@source*, *@precision*)

@tags supplies one or more valid language tags for the languages specified

Status Optional

Datatype 1–∞ occurrences of `data.language` separated by whitespace

Note This attribute should be supplied only if the element contains no <langKnown> children. Its values are language 'tags' as defined in RFC 4646 or its successor

Used by model.persTraitLike

May contain

core: p

linking: ab

namesdates: langKnown

Declaration

```
element langKnowledge
{
   att.global.attributes,
```

```
   att.datable.w3c.attributes,
   att.datable.iso.attributes,
   att.editLike.attributes,
   attribute tags { list { data.language+ } }?,
   ( model.pLike | langKnown+ )
}
```

Example

```
<langKnowledge  tags="en-GB fr">
  <p>British English and
     French</p>
</langKnowledge>
```

Example

```
<langKnowledge>
  <langKnown  tag="en-GB"  level="H">British English</langKnown>
  <langKnown  tag="fr"  level="M">French</langKnown>
</langKnowledge>
```

<langKnown> (language known) summarizes the state of a person's linguistic competence, i.e., knowledge of a single language.

Module namesdates — *13. Names, Dates, People, and Places*

Attributes att.datableatt.datable.w3c (*@period*, *@when*, *@notBefore*, *@notAfter*, *@from*, *@to*) att.datable.iso (*@when-iso*, *@notBefore-iso*, *@notAfter-iso*, *@from-iso*, *@to-iso*) att.editLike (*@cert*, *@resp*, *@evidence*, *@source*, *@precision*)

@tag supplies a valid language tag for the language concerned.

Status **Required**

Datatype `data.language`

Note The value for this attribute should be a language 'tag' as defined in BCP 47.

@level a code indicating the person's level of knowledge for this language

Status Optional

Datatype `data.word`

Used by langKnowledge

May contain

analysis: interp interpGrp span spanGrp

certainty: certainty respons

core: abbr address cb choice date distinct email emph expan foreign gap gloss index lb measure measureGrp mentioned milestone name note num pb ptr ref rs soCalled term time title

dictionaries: lang

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident tag val

textcrit: witDetail

transcr: addSpan am damageSpan delSpan ex fw handShift space subst

Declaration

```
element langKnown
{
   att.global.attributes,
   att.datable.w3c.attributes,
   att.datable.iso.attributes,
   att.editLike.attributes,
   attribute tag { data.language },
   attribute level { data.word }?,
   macro.phraseSeq.limited}
```

Example

```
<langKnown tag="en-GB" level="H">British English</langKnown>
<langKnown tag="fr" level="M">French</langKnown>
```

<langUsage>

(language usage) describes the languages, sublanguages, registers, dialects, etc. represented within a text.

Module header — *2. The TEI Header*

Attributes att.declarable (*@default*)

Used by model.profileDescPart

May contain

header: language

Declaration

```
element langUsage
{
   att.global.attributes,
   att.declarable.attributes,
   language+
}
```

Example

```
<langUsage>
  <language ident="fr-CA" usage="60">Québecois</language>
  <language ident="en-CA" usage="20">Canadian business English</language>
  <language ident="en-GB" usage="20">British English</language>
</langUsage>
```

<language> characterizes a single language or sublanguage used within a text.

Module header — *2. The TEI Header*

Attributes In addition to global attributes

@ident (identifier) Supplies a language code constructed as defined in BCP 47 which is used to identify the language documented by this element, and which is referenced by the global **xml:lang** attribute.

Status **Required**

Datatype `data.language`

@usage specifies the approximate percentage (by volume) of the text which uses this language.

Status Optional

Datatype `xsd:nonNegativeInteger { maxInclusive = "100" }`

Values a whole number between 0 and 100

Used by langUsage

May contain

analysis: interp interpGrp span spanGrp

certainty: certainty respons

core: abbr address cb choice date distinct email emph expan foreign gap gloss index lb measure measureGrp mentioned milestone name note num pb ptr ref rs soCalled term time title

dictionaries: lang

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident tag val

textcrit: witDetail

transcr: addSpan am damageSpan delSpan ex fw handShift space subst

Declaration

```
element language
{
   att.global.attributes,
   attribute ident { data.language },
   attribute usage { xsd:nonNegativeInteger { maxInclusive = "100" } }?,
   macro.phraseSeq.limited}
```

Example

```
<langUsage xml:lang="en-US">
  <language ident="en-US" usage="75">modern American English</language>
  <language ident="i-az-Arab" usage="20">Azerbaijani in Arabic script</language>
```

```
  <language ident="x-lap" usage="05">Pig Latin</language>
</langUsage>
```

Note Particularly for sublanguages, an informal prose characterization should be supplied as content for the element.

<layout> describes how text is laid out on the page, including information about any ruling, pricking, or other evidence of page-preparation techniques.

Module msdescription — *10. Manuscript Description*

Attributes In addition to global attributes

@columns specifies the number of columns per page
- *Status* Optional
- *Datatype* 1–2 occurrences of `data.count` separated by whitespace
- *Values* may be given as a pair of numbers (to indicate a range) or as a single number.

@ruledLines specifies the number of ruled lines per column
- *Status* Optional
- *Datatype* 1–2 occurrences of `data.count` separated by whitespace
- *Values* may be given as a pair of numbers (a range) or as a single number.

@writtenLines specifies the number of written lines per colum
- *Status* Optional
- *Datatype* 1–2 occurrences of `data.count` separated by whitespace
- *Values* may be given as a pair of numbers (a range), or as a single number.

Used by layoutDesc

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address bibl biblStruct binaryObject cb choice cit corr date del desc distinct email emph expan foreign gap gloss graphic hi index l label lb lg list listBibl measure measureGrp mentioned milestone name note num orig p pb ptr q quote ref reg rs said sic soCalled sp stage term time title unclear

dictionaries: lang oRef oVar pRef pVar

drama: camera caption castList move sound tech view

figures: figure formula table

gaiji: g

header: biblFull

iso-fs: fLib fs fvLib

linking: ab alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material msDesc origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat

geogName listNym listOrg listPerson listPlace nameLink offset orgName persName placeName region roleName settlement state surname

nets: eTree forest forestGrp graph tree

spoken: incident kinesic pause shift u vocal writing

tagdocs: att classSpec code eg egXML elementSpec gi ident listRef macroSpec moduleRef moduleSpec schemaSpec specDesc specGrp specGrpRef specList tag val

textcrit: app listWit witDetail

textstructure: floatingText

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element layout
{
   att.global.attributes,
   attribute columns { list { data.count, data.count? } }?,
   attribute ruledLines { list { data.count, data.count? } }?,
   attribute writtenLines { list { data.count, data.count? } }?,
   macro.specialPara}
```

Example

```
<layout columns="1" ruledLines="25 32">Most pages have between 25 and 32 long lines ruled in
lead.</layout>
```

Example

```
<layout columns="2" ruledLines="42">
  <p>2 columns of 42 lines ruled in ink, with central rule
     between the columns.</p>
</layout>
```

Example

```
<layout columns="1 2" writtenLines="40 50">
  <p>Some pages have 2 columns, with central rule
     between the columns; each column with between 40 and 50 lines of writing.</p>
</layout>
```

<layoutDesc> (layout description) collects the set of layout descriptions applicable to a manuscript.

Module msdescription — *10. Manuscript Description*

Attributes Global attributes only

Used by objectDesc

May contain

core: p

linking: ab

msdescription: layout

Declaration

```
element layoutDesc { att.global.attributes, ( model.pLike+ | layout+ ) }
```

Example

```
<layoutDesc>
  <p>Most pages have between 25 and 32 long lines ruled in lead.</p>
</layoutDesc>
```

Example

```
<layoutDesc>
  <layout  columns="2"  ruledLines="42">
    <p>
      <locus  from="f12r"  to="f15v"/>
         2 columns of 42 lines pricked and ruled in ink, with
         central rule between the columns.</p>
  </layout>
  <layout  columns="3">
    <p>
      <locus  from="f16"/>Prickings for three columns are visible.</p>
  </layout>
</layoutDesc>
```

<lb/> (line break) marks the start of a new (typographic) line in some edition or version of a text.

Module core — *3. Elements Available in All TEI Documents*

Attributes In addition to global attributes

@ed (edition) indicates the edition or version in which the line break is located at this point

Status Recommended

Datatype `data.code`

Values Any string of characters; usually a siglum conventionally used for the edition.

Used by model.milestoneLike

May contain Empty element

Declaration

```
element lb { att.global.attributes, attribute ed { data.code }?, empty }
```

Example Indicating typographical line breaks within metrical lines, where they occur at different places in different editions:

```
<l>Of Mans First Disobedience,<lb  ed="e2"/> and<lb  ed="e1"/> the Fruit</l>
<l>Of that Forbidden Tree, whose<lb  ed="e1 e2"/> mortal tast</l>
<l>Brought Death into the World,<lb  ed="e1"/> and all<lb  ed="e2"/> our woe,</l>
```

Example Indicating line structure of title page, display text, etc.:

```
<docTitle>
  <titlePart  type="main">THE
  <lb/>Pilgrim's Progress
  <lb/>FROM
```

```
  <lb/>THIS WORLD,
  <lb/>TO
  <lb/>That which is to come:
  </titlePart>
<!-- etc. -->
</docTitle>
```

Note On this element, the global **n** attribute indicates the number or other value associated with the line which follows the point of insertion of this <lb>. Encoders should adopt a clear and consistent policy as to whether the numbers associated with line breaks relate to the physical sequence number of the line within the page, or to some aspect of the logical structure of the text. By convention, <lb> elements should appear at the start of the line to which they refer.The <lb> tag is intended for making typographic line breaks in prose. It should be carefully distinguished from the <l> element, used to mark lines of verse.

<lbl> (label) contains a label for a form, example, translation, or other piece of information, e.g. abbreviation for, contraction of, literally, approximately, synonyms:, etc.

Module dictionaries — *9. Dictionaries*

Attributes att.lexicographic (*@expand*, *@norm*, *@split*, *@value*, *@orig*, *@location*, *@mergedIn*, *@opt*)

@type classifies the label using any convenient typology.

Status Optional

Datatype `data.enumerated`

Values any string of characters, such as *usage*, *sense_restriction*, etc.

Used by etym xr model.entryPart model.gramPart

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address bibl biblStruct binaryObject cb choice cit corr date del desc distinct email emph expan foreign gap gloss graphic hi index label lb list listBibl measure measureGrp mentioned milestone name note num orig pb ptr q quote ref reg rs said sic soCalled stage term time title unclear

dictionaries: lang oRef oVar pRef pVar

drama: camera caption castList move sound tech view

figures: figure formula table

gaiji: g

header: biblFull

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material msDesc origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName listNym listOrg listPerson listPlace nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att classSpec code eg egXML elementSpec gi ident listRef macroSpec moduleRef moduleSpec specDesc specGrp specGrpRef specList tag val

textcrit: app listWit witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element lbl
{
   att.global.attributes,
   att.lexicographic.attributes,
   attribute type { data.enumerated }?,
   macro.paraContent}
```

Example

```
<entry>
  <form type="abbrev">
    <orth>MTBF</orth>
  </form>
  <form type="full">
    <lbl>abbrev. for</lbl>
    <orth>mean time between failures</orth>
  </form>
</entry>
```

Note Labels specifically relating to usage should be tagged with the special-purpose <usg> element rather than with the generic<lbl> element.

<leaf> encodes the leaves (terminal nodes) of a tree.

Module nets — *19. Graphs, Networks, and Trees*

Attributes In addition to global attributes

@value provides a pointer to a feature structure or other analytic element.

Status Required when applicable

Datatype `data.pointer`

Values A valid identifier of a feature structure or other analytic element.

@parent provides the identifier of parent of a leaf.

Status Optional

Datatype `data.pointer`

Values The identifier of the parent node.

@follow provides an identifier of an element which this leaf follows.

Status Required when applicable

Datatype `data.pointer`

Values The identifier of another intermediate node or leaf of the tree.

Note If the tree is unordered or partially ordered, this attribute has the property of fixing the relative order of the leaf and the element which is the value of the attribute.

Used by tree

May contain

core: label

Declaration

```
element leaf
{
   att.global.attributes,
   attribute value { data.pointer }?,
   attribute parent { data.pointer }?,
   attribute follow { data.pointer }?,
   label?
}
```

Example

```
<leaf xml:id="peri1" parent="#n1">
  <label>periscope</label>
</leaf>
```

Note The in degree of a leaf is always 1, its out degree always 0.

<lem> (lemma) contains the lemma, or base text, of a textual variation.

Module textcrit — *12. Critical Apparatus*

Attributes att.textCritical (*@wit*, *@type*, *@cause*, *@varSeq*, *@resp*, *@hand*)

Used by app rdgGrp

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address bibl biblStruct binaryObject cb choice cit corr date del desc distinct email emph expan foreign gap gloss graphic hi index label lb list listBibl measure measureGrp mentioned milestone name note num orig pb ptr q quote ref reg rs said sic soCalled stage term time title unclear

dictionaries: lang oRef oVar pRef pVar

drama: camera caption castList move sound tech view

figures: figure formula table

gaiji: g

header: biblFull

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material msDesc origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName listNym listOrg listPerson listPlace nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att classSpec code eg egXML elementSpec gi ident listRef macroSpec moduleRef moduleSpec specDesc specGrp specGrpRef specList tag val

textcrit: app lacunaEnd lacunaStart listWit wit witDetail witEnd witStart

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element lem
{
   att.global.attributes,
   att.textCritical.attributes,
   (
      text
    | model.gLike    | model.phrase    | model.inter    | model.global    | model.rdgPart   )*
}
```

Example

```
<app>
  <lem  wit="#El #Hg">Experience</lem>
  <rdg  wit="#La"  type="substantive">Experiment</rdg>
  <rdg  wit="#Ra2"  type="substantive">Eryment</rdg>
</app>
```

Note The term lemma is used in text criticism to describe the reading in the text itself (as opposed to those in the apparatus); this usage is distinct from that of mathematics (where a lemma is a major step in a proof) and natural-language processing (where a lemma is the dictionary form associated with an inflected form in the running text).

<lg> (line group) contains a group of verse lines functioning as a formal unit, e.g. a stanza, refrain, verse paragraph, etc.

Module core — *3. Elements Available in All TEI Documents*

Attributes att.divLike (*@org*, *@sample*, *@part*) att.metrical (*@met*, *@real*, *@rhyme*) att.typed (*@type*, *@subtype*)

Used by lg sp model.divPart

May contain

analysis: interp interpGrp span spanGrp

certainty: certainty respons

core: cb gap head index l lb lg meeting milestone note pb

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp timeline

spoken: incident kinesic pause shift vocal writing

textcrit: witDetail

textstructure: argument byline closer dateline docAuthor docDate epigraph opener postscript salute signed trailer

transcr: addSpan damageSpan delSpan fw space

Declaration

```
element lg
{
   att.global.attributes,
   att.divLike.attributes,
   att.metrical.attributes,
   att.typed.attributes,
   (
      ( model.divTop | model.global )*,
      ( model.lLike | lg ),
      ( model.lLike | lg | model.global )*,
      ( ( model.divBottom ), model.global* )*
   )
}
```

Example

```
<lg  type="free">
  <l>Let me be my own fool</l>
  <l>of my own making, the sum of it</l>
</lg>
<lg  type="free">
  <l>is equivocal.</l>
  <l>One says of the drunken farmer:</l>
</lg>
<lg  type="free">
  <l>leave him lay off it. And this is</l>
  <l>the explanation.</l>
</lg>
```

Note contains verse lines or nested line groups only, possibly prefixed by a heading.

<link/> defines an association or hypertextual link among elements or passages, of some type not more precisely specifiable by other elements.

Module linking — *16. Linking, Segmentation, and Alignment*

Attributes att.pointing (*@type*, *@evaluate*)

@targets specifies the identifiers of the elements or passages to be linked or associated.

Status **Required**

Datatype 2–∞ occurrences of `data.pointer` separated by whitespace

Values two or more pointers (URIs), separated by whitespace

Used by linkGrp model.global.meta

May contain Empty element

Declaration

```
element link
{
   att.global.attributes,
   att.pointing.attributes,
   attribute targets { list { data.pointer, data.pointer+ } },
   empty
}
```

Example

```
<s  n="1">The state Supreme Court has refused to release
<rs  xml:id="R1">
   <rs  xml:id="R2">Rahway State Prison</rs> inmate</rs>
  <rs  xml:id="R3">James Scott</rs> on bail.</s>
<s  n="2">
  <rs  xml:id="R4">The fighter</rs> is serving 30-40 years for a 1975 armed
robbery conviction in <rs  xml:id="R5">the penitentiary</rs>.
</s>
<!-- ... -->
<linkGrp  type="periphrasis">
  <link  targets="#R1 #R3 #R4"/>
  <link  targets="#R2 #R5"/>
</linkGrp>
```

Note This element should only be used to encode associations not otherwise provided for by more specific elements.The location of this element within a document has no significance, unless it is included within a <linkGrp>, in which case it may inherit the value of the **type** attribute from the value given on the <linkGrp>.

<linkGrp> (link group) defines a collection of associations or hypertextual links.

Module linking — *16. Linking, Segmentation, and Alignment*

Attributes att.pointing.group (*@domains*, *@targFunc*) att.pointing (*@type*, *@evaluate*)

Used by model.global.meta

May contain

core: ptr

linking: link

Declaration

```
element linkGrp
{
   att.global.attributes,
   att.pointing.group.attributes,
   att.pointing.attributes,
   ( link | ptr )+
}
```

Example

```
<linkGrp  type="translation">
  <link  targets="#CCS1 #SW1"/>
  <link  targets="#CCS2 #SW2"/>
  <link  targets="#CCS #SW"/>
</linkGrp>
<div  type="volume"  xml:id="CCS">
  <p>
    <s  xml:id="CCS1">Longtemps, je me suis couché de bonne heure.</s>
    <s  xml:id="CCS2">Parfois, à peine ma bougie éteinte, mes yeux se fermaient si vite
        que je n'avais pas le temps de me dire : "Je m'endors."</s>
  </p>
<!-- ... -->
```

```
</div>
<div type="volume" xml:id="SW" xml:lang="en">
  <p>
    <s xml:id="SW1">For a long time I used to go to bed early.</s>
    <s xml:id="SW2">Sometimes, when I had put out my candle, my
        eyes would close so quickly that I had not even time to say
        "I'm going to sleep."</s>
  </p>
<!-- ... -->
</div>
```

Note May contain one or more <link> elements only, optionally with interspersed pointer elements.A web or link group is an administrative convenience, which should be used to collect a set of links together for any purpose, not simply to supply a default value for the **type** attribute.

<list> contains any sequence of items organized as a list.

Module core — *3. Elements Available in All TEI Documents*

Attributes In addition to global attributes

@type describes the form of the list.

Status Optional

Datatype `data.enumerated`

Suggested values include: **ordered** list items are numbered or lettered.

bulleted list items are marked with a bullet or other typographic device.

simple list items are not numbered or bulleted. [**Default**]

gloss each list item glosses some term or concept, which is given by a label element preceding the list item.

Note The formal syntax of the element declarations allows <label> tags to be omitted from lists tagged <**list type="gloss"**>; this is however a semantic error.

Used by keywords revisionDesc model.listLike

May contain

analysis: interp interpGrp span spanGrp

certainty: certainty respons

core: cb gap head headItem headLabel index item label lb meeting milestone note pb

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp timeline

spoken: incident kinesic pause shift vocal writing

textcrit: witDetail

textstructure: argument byline closer dateline docAuthor docDate epigraph opener postscript salute signed trailer

transcr: addSpan damageSpan delSpan fw space

Declaration

```
element list
{
   att.global.attributes,
```

```
   attribute type { "ordered" | "bulleted" | "simple" | "gloss" | xsd:Name }?,
   (
      ( ( model.divTop ) | ( model.global ) )*,
      (
         ( item, model.global* )+
       | (
            headLabel?,
            headItem?,
            ( label, model.global*, item, model.global* )+
         )
      ),
      ( ( model.divBottom ), model.global* )*
   )
}
```

Example

```
<list  type="ordered">
  <item>a butcher</item>
  <item>a baker</item>
  <item>a candlestick maker, with
  <list  type="bullets">
     <item>rings on his fingers</item>
     <item>bells on his toes</item>
   </list>
  </item>
</list>
```

Example The following example treats the short numbered clauses of Anglo-Saxon legal codes as lists of items. The text is from an ordinance of King Athelstan (924–939):

```
<div1  type="section">
  <head>Athelstan's Ordinance</head>
  <list  type="ordered">
    <item  n="1">Concerning thieves. First, that no thief is to be
        spared who is caught with the stolen goods, [if he is] over
        twelve years and [if the value of the goods is] over eightpence.
    <list  type="ordered">
       <item  n="1.1">And if anyone does spare one, he is to pay for the
              thief with his wergild – and the thief is to be no nearer a
              settlement on that account – or to clear himself by an oath of
              that amount.</item>
       <item  n="1.2">If, however, he [the thief] wishes to defend himself
              or to escape, he is not to be spared [whether younger or older
              than twelve].</item>
       <item  n="1.3">If a thief is put into prison, he is to be in prison
              40 days, and he may then be redeemed with 120 shillings; and the
              kindred are to stand surety for him that he will desist for ever.</item>
       <item  n="1.4">And if he steals after that, they are to pay for him
              with his wergild, or to bring him back there.</item>
       <item  n="1.5">And if he steals after that, they are to pay
              for him with his wergild, whether to the king or to him
              to whom it rightly belongs; and everyone of those who
              supported him is to pay 120 shillings to the king as a fine.</item>
     </list>
    </item>
    <item  n="2">Concerning lordless men. And we pronounced about these
```

```
            lordless men, from whom no justice can be obtained, that one
            should order their kindred to fetch back such a person to justice
            and to find him a lord in public meeting.
        <list type="ordered">
           <item n="2.1">And if they then will not, or cannot, produce him on
                  that appointed day, he is then to be a fugitive afterwards, and
                  he who encounters him is to strike him down as a thief.</item>
           <item n="2.2">And he who harbours him after that, is to pay for him
                  with his wergild or to clear himself by an oath of that amount.</item>
         </list>
        </item>
        <item n="3">Concerning the refusal of justice. The lord who
            refuses justice and upholds his guilty man, so that the king is
            appealed to, is to repay the value of the goods and 120 shillings
            to the king; and he who appeals to the king before he demands
            justice as often as he ought, is to pay the same fine as the
            other would have done, if he had refused him justice.
        <list type="ordered">
           <item n="3.1">And the lord who is an accessory to a theft by his
                  slave, and it becomes known about him, is to forfeit the slave
                  and be liable to his wergild on the first occasionp if he does it
                  more often, he is to be liable to pay all that he owns.</item>
           <item n="3.2">And likewise any of the king's treasurers or of our
                  reeves, who has been an accessory of thieves who have committed
                  theft, is to liable to the same.</item>
         </list>
        </item>
        <item n="4">Concerning treachery to a lord. And we have pronounced
            concerning treachery to a lord, that he [who is accused] is to
            forfeit his life if he cannot deny it or is afterwards convicted
            at the three-fold ordeal.</item>
      </list>
    </div1>
```

Note that nested lists have been used so the tagging mirrors the structure indicated by the two-level numbering of the clauses. The clauses could have been treated as a one-level list with irregular numbering, if desired.

Example

```
<p>These decrees, most blessed Pope Hadrian, we propounded in the
public council ... and they confirmed them in our hand in your
stead with the sign of the Holy Cross, and afterwards inscribed
with a careful pen on the paper of this page, affixing thus the
sign of the Holy Cross.
<list type="simple">
   <item>I, Eanbald, by the grace of God archbishop of the holy
       church of York, have subscribed to the pious and catholic
       validity of this document with the sign of the Holy Cross.</item>
   <item>I, Ælfwold, king of the people across the Humber,
       consenting have subscribed with the sign of the Holy Cross.</item>
   <item>I, Tilberht, prelate of the church of Hexham, rejoicing
       have subscribed with the sign of the Holy Cross.</item>
   <item>I, Higbald, bishop of the church of Lindisfarne, obeying
       have subscribed with the sign of the Holy Cross.</item>
   <item>I, Ethelbert, bishop of Candida Casa, suppliant, have
       subscribed with thef sign of the Holy Cross.</item>
```

```
    <item>I, Ealdwulf, bishop of the church of Mayo, have subscribed
        with devout will.</item>
    <item>I, Æthelwine, bishop, have subscribed through
        delegates.</item>
    <item>I, Sicga, patrician, have subscribed with serene mind
        with the sign of the Holy Cross.</item>
  </list>
</p>
```

Note May contain an optional heading followed by a series of items, or a series of label and item pairs, the latter being optionally preceded by one or two specialized headings.

<listBibl> (citation list) contains a list of bibliographic citations of any kind.

Module core — *3. Elements Available in All TEI Documents*

Attributes att.declarable (*@default*) att.typed (*@type*, *@subtype*)

Used by additional listBibl msItemStruct model.listLike model.msItemPart

May contain

core: bibl biblStruct head listBibl

header: biblFull

msdescription: msDesc

Declaration

```
element listBibl
{
   att.global.attributes,
   att.declarable.attributes,
   att.typed.attributes,
   ( model.headLike*, ( model.biblLike | listBibl )+ )
}
```

Example

```
<listBibl>
  <head>Works consulted</head>
  <bibl>Blain, Clements and Grundy: Feminist Companion to
     Literature in English (Yale, 1990)
  </bibl>
  <biblStruct>
    <analytic>
      <title>The Interesting story of the Children in the Wood</title>
    </analytic>
    <monogr>
      <title>The Penny Histories</title>
      <author>Victor E Neuberg</author>
      <imprint>
        <publisher>OUP</publisher>
        <date>1968</date>
      </imprint>
    </monogr>
  </biblStruct>
</listBibl>
```

<listNym> (list of canonical names) contains a list of nyms, that is, standardized names for any thing.

Module namesdates — *13. Names, Dates, People, and Places*

Attributes att.typed (*@type*, *@subtype*) att.declarable (*@default*)

Used by listNym model.listLike

May contain

core: head

namesdates: listNym nym relation relationGrp

Declaration

```
element listNym
{
   att.global.attributes,
   att.typed.attributes,
   att.declarable.attributes,
   ( model.headLike*, ( nym | listNym )+, ( relationGrp | relation )* )
}
```

Example

```
<listNym  type="floral">
  <nym  xml:id="ROSE">
    <form>Rose</form>
  </nym>
  <nym  xml:id="DAISY">
    <form>Daisy</form>
    <etym>Contraction of <mentioned>day's eye</mentioned>
    </etym>
  </nym>
  <nym  xml:id="HTHR">
    <form>Heather</form>
  </nym>
</listNym>
```

Note The type attribute may be used to distinguish lists of names of a particular type if convenient.

<listOrg> (list of organizations) contains a list of elements, each of which provides information about an identifiable organization.

Module namesdates — *13. Names, Dates, People, and Places*

Attributes att.typed (*@type*, *@subtype*) att.declarable (*@default*)

Used by listOrg model.listLike

May contain

core: head

namesdates: listOrg org relation relationGrp

Declaration

```
element listOrg
{
   att.global.attributes,
```

```
   att.typed.attributes,
   att.declarable.attributes,
   ( model.headLike*, ( org | listOrg )+, ( relationGrp | relation )* )
}
```

Example

```
<listOrg>
  <head>Libyans</head>
  <org>
    <orgName>Adyrmachidae</orgName>
    <desc>These people have, in most points, the same customs as the
        Egyptians, but use the costume of the Libyans. Their women wear on
        each leg a ring made of bronze [...] </desc>
  </org>
  <org>
    <orgName>Nasamonians</orgName>
    <desc>In summer they leave their flocks and herds upon the
        sea-shore, and go up the country to a place called Augila, where
        they gather the dates from the palms [...]</desc>
  </org>
  <org>
    <orgName>Garamantians</orgName>
    <desc>[...] avoid all society or intercourse with their fellow-men,
        have no weapon of war, and do not know how to defend
        themselves. [...]</desc>
<!-- ... -->
  </org>
</listOrg>
```

Source: [98]

Note The type attribute may be used to distinguish lists of organizations of a particular type if convenient.

<listPerson>

<listPerson> (list of persons) contains a list of descriptions, each of which provides information about an identifiable person or a group of people, for example the participants in a language interaction, or the people referred to in a historical source.

Module namesdates — *13. Names, Dates, People, and Places*

Attributes att.typed (*@type*, *@subtype*) att.declarable (*@default*)

Used by listPerson particDesc model.listLike

May contain

core: head

namesdates: listPerson org person personGrp relation relationGrp

Declaration

```
element listPerson
{
   att.global.attributes,
   att.typed.attributes,
   att.declarable.attributes,
   (
      model.headLike*,
```

```
      ( model.personLike | listPerson )+,
      ( relation | relationGrp )*
   )
}
```

Example

```
<listPerson type="respondents">
  <personGrp xml:id="PXXX"/>
  <person xml:id="P1234" sex="2" age="mid"/>
  <person xml:id="P4332" sex="1" age="mid"/>
  <relationGrp>
    <relation type="personal" name="spouse" mutual="#P1234 #P4332"/>
  </relationGrp>
</listPerson>
```

Note The type attribute may be used to distinguish lists of people of a particular type if convenient.

<listPlace> (list of places) contains a list of places, optionally followed by a list of relationships (other than containment) defined amongst them.

Module namesdates — *13. Names, Dates, People, and Places*

Attributes att.typed (*@type*, *@subtype*) att.declarable (*@default*)

Used by listPlace place model.listLike

May contain

core: head

namesdates: listPlace place relation relationGrp

Declaration

```
element listPlace
{
   att.global.attributes,
   att.typed.attributes,
   att.declarable.attributes,
   (
      model.headLike*,
      ( model.placeLike | listPlace )+,
      ( relationGrp | relation )*
   )
}
```

Example

```
<listPlace type="offshoreIslands">
  <place>
    <placeName>La roche qui pleure</placeName>
  </place>
  <place>
    <placeName>Ile aux cerfs</placeName>
  </place>
</listPlace>
```

<listRef> (list of references) supplies a list of significant references to places where this element is discussed, in the current document or elsewhere.

Module tagdocs — *22. Documentation Elements*

Attributes Global attributes only

Used by classSpec elementSpec macroSpec moduleSpec model.oddDecl

May contain

core: ptr

Declaration

```
element listRef { att.global.attributes, ptr+ }
```

Example

```
<listRef>
  <ptr target="#ddc12"/>
</listRef>
```

<listWit> (witness list) lists definitions for all the witnesses referred to by a critical apparatus, optionally grouped hierarchically.

Module textcrit — *12. Critical Apparatus*

Attributes Global attributes only

Used by listWit model.listLike

May contain

core: head

textcrit: listWit witness

Declaration

```
element listWit
{
   att.global.attributes,
   ( model.headLike?, ( witness | listWit )+ )
}
```

Example

```
<listWit>
  <witness xml:id="HL26">Ellesmere, Huntingdon Library 26.C.9</witness>
  <witness xml:id="PN392">Hengwrt, National Library of Wales,
    Aberystwyth, Peniarth 392D</witness>
  <witness xml:id="RP149">Bodleian Library Rawlinson Poetic 149
    (see further <ptr target="#MSRP149"/>)</witness>
</listWit>
```

Note May contain a series of <witness> or <listWit> elements. The provision of a <listWit> element simplifies the automatic processing of the apparatus, e.g. the reconstruction of the readings for all witnesses from an exhaustive apparatus.Situations commonly arise where there are many more or less fragmentary witnesses, such that there may be quite distinct groups of witnesses for different parts of a text or collection of texts. Such groups may be given separately, or nested within a

single <listWit> element at the beginning of the file listing all the witnesses, partial and complete, for the text, with the attestation of fragmentary witnesses indicated within the apparatus by use of the <witStart> and <witEnd> elements described in section *12.1.5. Fragmentary Witnesses*.Note however that a given witness can only be defined once, and can therefore only appear within a single <listWit> element.

<localName> (locally-defined property name) contains a locally defined name for some property.

Module gaiji — *5. Representation of Non-standard Characters and Glyphs*

Attributes Global attributes only

Used by charProp

May contain Character data only

Declaration `element localName { att.global.attributes, text }`

Example

```
<localName>daikanwa</localName>
<localName>entity</localName>
```

Note No definitive list of local names is proposed. However, the name entity is recommended as a means of naming the property identifying the recommended character entity name for this character or glyph.

<locale> contains a brief informal description of the kind of place concerned, for example: a room, a restaurant, a park bench, etc.

Module corpus — *15. Language Corpora*

Attributes Global attributes only

Used by model.settingPart

May contain

analysis: interp interpGrp span spanGrp

certainty: certainty respons

core: abbr address cb choice date distinct email emph expan foreign gap gloss index lb measure measureGrp mentioned milestone name note num pb ptr ref rs soCalled term time title

dictionaries: lang

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident tag val

textcrit: witDetail

transcr: addSpan am damageSpan delSpan ex fw handShift space subst

Declaration

```
element locale { att.global.attributes, macro.phraseSeq.limited }
```

Example

```
<locale>a fashionable restaurant</locale>
```

<location> defines the location of a place as a set of geographical coordinates, in terms of a other named geo-political entities, or as an address.

Module namesdates — *13. Names, Dates, People, and Places*

Attributes att.typed (*@type*, *@subtype*) att.datableatt.datable.w3c (*@period*, *@when*, *@notBefore*, *@notAfter*, *@from*, *@to*) att.datable.iso (*@when-iso*, *@notBefore-iso*, *@notAfter-iso*, *@from-iso*, *@to-iso*) att.editLike (*@cert*, *@resp*, *@evidence*, *@source*, *@precision*)

Used by model.placeTraitLike

May contain

core: address bibl biblStruct desc email label measure measureGrp note num

header: biblFull

msdescription: depth height msDesc width

namesdates: affiliation bloc country district geo geogFeat geogName offset placeName region settlement

textcrit: witDetail

Declaration

```
element location
{
   att.global.attributes,
   att.typed.attributes,
   att.datable.w3c.attributes,
   att.datable.iso.attributes,
   att.editLike.attributes,
   (
      model.labelLike    | model.placeNamePart    | model.offsetLike    | model.measureLike    | model.addressLi
}
```

Example

```
<place>
  <placeName>Abbey Dore</placeName>
  <location>
    <geo>51.969604 -2.893146</geo>
  </location>
</place>
```

Example

```
<place  type="building">
  <placeName>Brasserie Georges</placeName>
```

```
  <location>
    <country key="FR"/>
    <settlement type="city">Lyon</settlement>
    <district type="arrondissement">Perrache</district>
    <placeName type="street">Rue de la Charité</placeName>
  </location>
</place>
```

Example

```
<place type="imaginary">
  <placeName>Atlantis</placeName>
  <location>
    <offset>beyond</offset>
    <placeName>The Pillars of <persName>Hercules</persName>
    </placeName>
  </location>
</place>
```

<locus> defines a location within a manuscript or manuscript part, usually as a (possibly discontinuous) sequence of folio references.

Module msdescription — *10. Manuscript Description*

Attributes In addition to global attributes

@scheme identifies the foliation scheme in terms of which the location is being specified.

Status Optional

Datatype `data.pointer`

Values A pointer to some <foliation> element which defines the foliation scheme used, or an external link to some equivalent resource.

@from specifies the starting point of the location in a normalized form.

Status Optional

Datatype `data.word`

Values typically this will be a page number

@to specifies the end-point of the location in a normalized form.

Status Optional

Datatype `data.word`

Values typically this will be a page number

@target supplies a link to one or more transcriptions of the specified range of folios.

Status Optional

Datatype 1–∞ occurrences of `data.pointer` separated by whitespace

Used by msItem msItemStruct model.pPart.msdesc

May contain

gaiji: g

Declaration

```
element locus
{
   att.global.attributes,
```

```
  attribute scheme { data.pointer }?,
  attribute from { data.word }?,
  attribute to { data.word }?,
  attribute target { list { data.pointer+ } }?,
  macro.xtext}
```

Example

```
<!-- within ms description --><msItem  n="1">
  <locus  target="#F1r #F1v #F2r">ff. 1r-2r</locus>
  <author>Ben Jonson</author>
  <title>Ode to himself</title>
  <rubric  rend="italics"> An Ode<lb/> to him selfe.</rubric>
  <incipit>Com leaue the loathed stage</incipit>
  <explicit>And see his chariot triumph ore his wayne.</explicit>
  <bibl>
    <name>Beal</name>, <title>Index 1450-1625</title>, JnB 380</bibl>
</msItem>
<!-- within transcription ... -->
<pb  xml:id="F1r"/>
<!-- ... -->
<pb  xml:id="F1v"/>
<!-- ... -->
<pb  xml:id="F2r"/>
<!-- ... -->
```

Example The **facs** attribute is available globally when the transcr module is included in a schema. It may be used to point directly to an image file, as in the following example:

```
<msItem>
  <locus
     facs="images/08v.jpg images/09r.jpg images/09v.jpg images/10r.jpg images/10v.jpg">fols.
8v-10v</locus>
  <title>Birds Praise of Love</title>
  <bibl>
    <title>IMEV</title>
    <biblScope>1506</biblScope>
  </bibl>
</msItem>
```

Note The **target** attribute should only be used to point to elements that contain or indicate a transcription of the locus being described, as in the first example above. To associate a <locus> element with a page image or other comparable representation, the global **facs** attribute should be used instead, as shown in the second example. Use of the **target** attribute to indicate an image is strongly deprecated. The **facs** attribute may be used to indicate one or more image files, as above, or alternatively it may point to one or more appropriate XML elements, such as the <surface>, <zone> element, <graphic>, or <binaryObject> elements.

<m> (morpheme) represents a grammatical morpheme.

Module analysis — *17. Simple Analytic Mechanisms*

Attributes att.segLike (*@type*, *@function*, *@part*) att.metrical (*@met*, *@real*, *@rhyme*)

@baseForm identifies the morpheme's base form.

Status Optional

Datatype `data.word`

Values a string of characters representing the spelling of the morpheme's base form.

Used by model.segLike

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: cb gap index lb milestone note pb

gaiji: g

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

spoken: incident kinesic pause shift vocal writing

textcrit: witDetail

transcr: addSpan damageSpan delSpan fw space

Declaration

```
element m
{
   att.global.attributes,
   att.segLike.attributes,
   att.metrical.attributes,
   attribute baseForm { data.word }?,
   ( text | model.gLike | model.segLike | model.global )*
}
```

Example

```
<w type="adjective">
  <w type="noun">
    <m type="prefix" baseForm="con">com</m>
    <m type="root">fort</m>
  </w>
  <m type="suffix">able</m>
</w>
```

Note May contain character data, <seg>, and <c> elements only.

<macroSpec> (macro specification) documents the function and implementation of a pattern.

Module tagdocs — *22. Documentation Elements*

Attributes att.identified (*@ident*, *@predeclare*, *@module*, *@mode*)

@type indicates which type of entity should be generated, when an ODD processor is generating a module using XML DTD syntax.

Status Optional

Legal values are: **pe** (parameter entity)

dt (datatype entity)

Used by model.oddDecl

May contain

core: desc gloss

tagdocs: altIdent content equiv exemplum listRef remarks stringVal

Declaration

```
element macroSpec
{
   att.global.attributes,
   att.identified.attributes,
   attribute type { "pe" | "dt" }?,
   (
      model.glossLike*,
      ( stringVal | content )+,
      exemplum*,
      remarks*,
      listRef*
   )
}
```

Example

```
<macroSpec module="tei" type="pe" ident="macro.phraseSeq">
  <content>
    <rng:zeroOrMore>
      <rng:choice>
        <rng:text/>
        <rng:ref name="model.gLike"/>
        <rng:ref name="model.phrase"/>
        <rng:ref name="model.global"/>
      </rng:choice>
    </rng:zeroOrMore>
  </content>
</macroSpec>
```

<mapping>

<mapping> (character mapping) contains one or more characters which are related to the parent character or glyph in some respect, as specified by the **type** attribute.

Module gaiji — *5. Representation of Non-standard Characters and Glyphs*

Attributes att.typed (*@type*, *@subtype*)

Used by char glyph

May contain

gaiji: g

Declaration

```
element mapping { att.global.attributes, att.typed.attributes, macro.xtext }
```

Example

```
<mapping type="modern">r</mapping>
<mapping type="standard">人</mapping>
```

Note Suggested values for the **type** attribute include `exact` for exact equivalences, `uppercase` for uppercase equivalences, `lowercase` for lowercase equivalences, and `simplified` for simplified

characters. The <g> elements contained by this element can point to either another <char> or <glyph>element or contain a character that is intended to be the target of this mapping.

<material> contains a word or phrase describing the material of which a manuscript (or part of a manuscript) is composed.

Module msdescription — *10. Manuscript Description*

Attributes Global attributes only

Used by model.pPart.msdesc

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address binaryObject cb choice corr date del distinct email emph expan foreign gap gloss graphic hi index lb measure measureGrp mentioned milestone name note num orig pb ptr ref reg rs sic soCalled term time title unclear

dictionaries: lang oRef oVar pRef pVar

figures: formula

gaiji: g

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident specDesc specList tag val

textcrit: app witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element material { att.global.attributes, macro.phraseSeq }
```

Example

```
<physDesc><p>
<material>Parchment</material> leaves with a
<material>sharkskin</material> binding.</p></physDesc>
```

<measure> contains a word or phrase referring to some quantity of an object or commodity, usually comprising a number, a unit, and a commodity name.

Module core — *3. Elements Available in All TEI Documents*

Attributes att.measurement (*@unit*, *@quantity*, *@commodity*)

@type specifies the type of measurement in any convenient typology.

Status Mandatory when applicable

Datatype `data.enumerated`

Used by model.measureLike

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address binaryObject cb choice corr date del distinct email emph expan foreign gap gloss graphic hi index lb measure measureGrp mentioned milestone name note num orig pb ptr ref reg rs sic soCalled term time title unclear

dictionaries: lang oRef oVar pRef pVar

figures: formula

gaiji: g

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident specDesc specList tag val

textcrit: app witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element measure
{
   att.global.attributes,
   att.measurement.attributes,
   attribute type { data.enumerated }?,
   macro.phraseSeq}
```

Example

```
<measure type="weight">
  <num>2</num> pounds of flesh
</measure>
<measure type="currency">£10-11-6d</measure>
<measure type="area">2 merks of old extent</measure>
```

Example

```
<measure quantity="40" unit="hogshead" commodity="rum">2 score hh rum</measure>
<measure quantity="12" unit="count" commodity="roses">1 doz. roses</measure>
<measure quantity="1" unit="count" commodity="tulips">a yellow tulip</measure>
```

<measureGrp> (measure group) contains a group of dimensional specifications which relate to the same object, for example the height and width of a manuscript page.

Module core — *3. Elements Available in All TEI Documents*

Attributes att.measurement (*@unit*, *@quantity*, *@commodity*) att.typed (*@type*, *@subtype*)

Used by model.measureLike

May contain

core: measure measureGrp num

gaiji: g

msdescription: depth height width

namesdates: geo

Declaration

```
element measureGrp
{
   att.global.attributes,
   att.measurement.attributes,
   att.typed.attributes,
   ( text | model.gLike | model.measureLike )*
}
```

Example

```
<measureGrp type="leaves" unit="mm">
  <height scope="range">157-160</height>
  <width quantity="105"/>
</measureGrp>
<measureGrp type="ruledArea" unit="mm">
  <height scope="most" quantity="90"/>
  <width scope="most" quantity="48"/>
</measureGrp>
<measureGrp type="box" unit="in">
  <height quantity="12"/>
  <width quantity="10"/>
  <depth quantity="6"/>
</measureGrp>
```

<meeting> contains the formalized descriptive title for a meeting or conference, for use in a bibliographic description for an item derived from such a meeting, or as a heading or preamble to publications emanating from it.

Module core — *3. Elements Available in All TEI Documents*

Attributes Global attributes only

Used by monogr model.divWrapper model.biblPart

May contain

core: abbr address bibl biblStruct choice cit date desc distinct email emph expan foreign gloss label list listBibl measure measureGrp mentioned name num ptr q quote ref rs said soCalled stage term time title

dictionaries: lang

drama: camera caption castList move sound tech view

figures: figure table

header: biblFull

msdescription: catchwords depth dimensions height heraldry locus material msDesc origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName listNym listOrg listPerson listPlace nameLink offset orgName persName placeName region roleName settlement state surname

tagdocs: att classSpec code eg egXML elementSpec gi ident listRef macroSpec moduleRef moduleSpec specGrp specGrpRef tag val

textcrit: listWit

transcr: am ex handShift subst

Declaration

```
element meeting { att.global.attributes, macro.limitedContent }
```

Example

```
<div>
  <meeting>Ninth International Conference on Middle High German
     Textual Criticism, Aachen, June 1998.</meeting>
  <list  type="attendance">
    <head>List
        of
        Participants</head>
    <item>
      <persName>...</persName>
    </item>
    <item>
      <persName>...</persName>
    </item>
<!--...-->
  </list>
  <p>...</p>
</div>
```

<memberOf> specifies class membership of the parent element or class.

Module tagdocs — *22. Documentation Elements*

Attributes In addition to global attributes

@key specifies the identifier for a class of which the documented element or class is a member or subclass

Status Optional

Datatype `data.name`

@mode specifies the effect of this declaration on its parent module.
Status Optional
Legal values are: **add** this declaration is added to the current definitions [**Default**]
delete this declaration and all of its children are removed from the current setup

Used by classes

May contain

gaiji: g

Declaration

```
element memberOf
{
   att.global.attributes,
   attribute key { data.name }?,
   attribute mode { "add" | "delete" }?,
   macro.xtext}
```

Example

```
<memberOf key="model.divLike"/>
<memberOf key="att.identified"/>
```

This element will appear in any content model which references model.divLike, and will have attributes defined in att.identified (in addition to any defined explicitly for this element).

Note Elements or classes which are members of multiple (unrelated) classes will have more than one <memberOf> element, grouped by a <classes> element. If an element is a member of a class C1, which is itself a subclass of a class C2, there is no need to state this, other than in the documentation for class C1. Any additional comment or explanation of the class membership may be provided as content for this element.

<mentioned> marks words or phrases mentioned, not used.

Module core — *3. Elements Available in All TEI Documents*

Attributes Global attributes only

Used by model.emphLike

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address binaryObject cb choice corr date del distinct email emph expan foreign gap gloss graphic hi index lb measure measureGrp mentioned milestone name note num orig pb ptr ref reg rs sic soCalled term time title unclear

dictionaries: lang oRef oVar pRef pVar

figures: formula

gaiji: g

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident specDesc specList tag val

textcrit: app witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element mentioned { att.global.attributes, macro.phraseSeq }
```

Example

```
The weight of the evidence clearly
demonstrates, as noted, that the systemic change from
<mentioned>creation</mentioned> to <mentioned>intelligent
design</mentioned> occurred sometime in 1987, after the Supreme
Court's important Edwards decision. This compelling evidence strongly
supports Plaintiffs' assertion that ID is creationism
re-labeled.
```

Source: [201]

<metDecl> (metrical notation declaration) documents the notation employed to represent a metrical pattern when this is specified as the value of a **met**, **real**, or **rhyme** attribute on any structural element of a metrical text (e.g. <lg>, <l>, or <seg>).

Module verse — *6. Verse*

Attributes att.declarable (*@default*)

@type indicates whether the notation conveys the abstract metrical form, its actual prosodic realization, or the rhyme scheme, or some combination thereof.

Status Mandatory when applicable

Datatype 1–3 occurrences of `data.enumerated` separated by whitespace

Values are: **met** (met attribute) declaration applies to the abstract metrical form recorded on the **met** attribute

real (real attribute) declaration applies to the actual realization of the conventional metrical structure recorded on the **real** attribute

rhyme (rhyme attribute) declaration applies to the rhyme scheme recorded on the **rhyme** attribute

Note By default, the <metDecl> element documents the notation used for metrical pattern and realization. It may also be used to document the notation used for rhyme scheme information; if not otherwise documented, the rhyme scheme notation defaults to the traditional 'abab' notation.

@pattern (regular expression pattern) specifies a regular expression defining any value that is legal for this notation.

Status Optional

Datatype `data.pattern`

Values the value must be a valid regular expression per the World Wide Web Consortium's *XML Schema Part 2: Datatypes Second Edition*, Appendix F

Used by model.encodingPart

May contain

core: note p

linking: ab

textcrit: witDetail

verse: metSym

Declaration

```
element metDecl
{
   att.global.attributes,
   att.declarable.attributes,
   attribute type
   {
      list { data.enumerated, data.enumerated?, data.enumerated? }
   }?,
   attribute pattern { data.pattern }?,
   ( ( model.pLike | model.noteLike )+ | metSym+ )
}
```

Example

```
<metDecl xml:id="ip" type="met" pattern="((SU|US)USUSUSUS/)">
  <metSym value="S">stressed syllable</metSym>
  <metSym value="U">unstressed syllable</metSym>
  <metSym value="/">metrical line boundary</metSym>
</metDecl>
```

This example is intended for the far more restricted case typified by the Shakespearean iambic pentameter. Only metrical patterns containing exactly ten syllables, alternately stressed and unstressed, (except for the first two which may be in either order) to each metrical line can be expressed using this notation.

Note The encoder may choose whether to define the notation formally or informally. However, the two methods may not be mixed. That is, <metDecl> may contain either a sequence of <metSym> elements or, alternately, a series of paragraphs or other components. If the **pattern** attribute is specified and <metSym> elements are used, then all the codes appearing within the **pattern** attribute should be documented.Only usable within the header if the verse module is used.

<metSym>

(metrical notation symbol) documents the intended significance of a particular character or character sequence within a metrical notation, either explicitly or in terms of other symbol elements in the same metDecl.

Module verse — *6. Verse*

Attributes In addition to global attributes

@value specifies the character or character sequence being documented.

Status **Required**

Datatype 1–∞ occurrences of `data.word` separated by whitespace

Values any available character or character sequence.

@terminal specifies whether the symbol is defined in terms of other symbols (**terminal** is set to **false**) or in prose (**terminal** is set to **true**).

Status Mandatory when applicable

Datatype `data.truthValue`

Note The value **true** indicates that the element contains a prose definition of its meaning; the value **false** indicates that the element contains a definition of its meaning given using symbols defined elsewhere in the same <metDecl> element.

Used by metDecl

May contain

analysis: interp interpGrp span spanGrp

certainty: certainty respons

core: abbr address cb choice date distinct email emph expan foreign gap gloss index lb measure measureGrp mentioned milestone name note num pb ptr ref rs soCalled term time title

dictionaries: lang

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident tag val

textcrit: witDetail

transcr: addSpan am damageSpan delSpan ex fw handShift space subst

Declaration

```
element metSym
{
   att.global.attributes,
   attribute value { list { data.word+ } },
   attribute terminal { data.truthValue }?,
   macro.phraseSeq.limited}
```

Example

```
<metSym  value="x">a stressed syllable</metSym>
<metSym  value="o">an unstressed syllable</metSym>
<metSym  value="A"  terminal="false">xoo</metSym>
```

<milestone/> marks a boundary point separating any kind of section of a text, typically but not necessarily indicating a point at which some part of a standard reference system changes, where the change is not represented by a structural element.

Module core — *3. Elements Available in All TEI Documents*

Attributes In addition to global attributes

@ed (edition) indicates the edition or version to which the milestone applies.

Status Recommended

Datatype `data.code`

Values Any string of characters; usually a siglum conventionally used for the edition.

@unit provides a conventional name for the kind of section changing at this milestone.

Status **Required**

Datatype `data.enumerated`

Suggested values include: **page** physical page breaks (synonymous with the <pb> element).

column column breaks.

line line breaks (synonymous with the <lb> element).

book any units termed book, liber, etc.

poem individual poems in a collection.

canto cantos or other major sections of a poem.

speaker changes of speaker or narrator.

stanza stanzas within a poem, book, or canto.

act acts within a play.

scene scenes within a play or act.

section sections of any kind.

absent passages not present in the reference edition.

unnumbered passages present in the text, but not to be included as part of the reference.

Note If the milestone marks the beginning of a piece of text not present in the reference edition, the special value absent may be used as the value of **unit**. The normal interpretation is that the reference edition does not contain the text which follows, until the next <milestone> tag for the edition in question is encountered.In addition to the values suggested, other terms may be appropriate (e.g. Stephanus for the Stephanus numbers in Plato).

Used by model.milestoneLike

May contain Empty element

Declaration

```
element milestone
{
   att.global.attributes,
   attribute ed { data.code }?,
   attribute unit
   {
```

```
      "page"
    | "column"
    | "line"
    | "book"
    | "poem"
    | "canto"
    | "speaker"
    | "stanza"
    | "act"
    | "scene"
    | "section"
    | "absent"
    | "unnumbered"
    | xsd:Name
   },
   empty
}
```

Example

```
<milestone n="23" ed="La" unit="Dreissiger"/>
...
<milestone n="24" ed="AV" unit="verse"/>
...
```

Note For this element, the global **n** attribute indicates the new number or other value for the unit which changes at this milestone. The special value *unnumbered* should be used in passages which fall outside the normal numbering scheme, such as chapter or other headings, poem numbers or titles, etc. The order in which milestone elements are given at a given point is not normally significant.

<moduleRef> (module reference) references a module which is to be incorporated into a schema.

Module tagdocs — *22. Documentation Elements*

Attributes In addition to global attributes

@key the name of a TEI module

Status Optional

Datatype `xsd:NCName`

@url (uniform resource locator) refers to a non-TEI module of RELAX NG code by external location

Status Optional

Datatype `data.pointer`

Used by schemaSpec model.oddRef

May contain

tagdocs: content

Declaration

```
element moduleRef
{
   att.global.attributes,
   ( attribute key { xsd:NCName }? | attribute url { data.pointer }? ),
```

```
   content?
}
```

<sch:pattern name="testschemapattern"> <sch:rule context="tei:moduleRef">
<sch:report test="* and @key">child elements of moduleRef are only allowed when an external module is being loaded </sch:report> </sch:rule> </sch:pattern>

Example

```
<moduleRef key="linking"/>
```

This embeds the linking module.

Note Modules are identified by the name supplied as value for the **ident** attribute on the <moduleSpec> element in which they are declared. A URI may also be supplied in the case of a non-TEI module, and this is expected to be written as a RELAX NG schema. The effect of this element is to make all the declarations contained by the referenced module available to the schema being compiled.

<moduleSpec> (module specification) documents the structure, content, and purpose of a single module, i.e. a named and externally visible group of declarations.

Module tagdocs — *22. Documentation Elements*

Attributes att.identified (*@ident*, *@predeclare*, *@module*, *@mode*)

@type type of module to be generated

Status Optional

Values A closed set of keywords yet to be defined

Used by model.oddDecl

May contain

core: desc gloss

tagdocs: altIdent equiv exemplum listRef remarks

Declaration

```
element moduleSpec
{
   att.global.attributes,
   att.identified.attributes,
   attribute type { text }?,
   ( model.glossLike*, exemplum*, remarks?, listRef* )
}
```

Example

```
<moduleSpec ident="namesdates">
  <altIdent type="FPI">Names and Dates</altIdent>
  <desc>Additional elements for names and dates</desc>
</moduleSpec>
```

<monogr> (monographic level) contains bibliographic elements describing an item (e.g. a book or journal) published as an independent item (i.e. as a separate physical object).

Module core — *3. Elements Available in All TEI Documents*

Attributes Global attributes only

Used by biblStruct

May contain

core: author biblScope editor imprint meeting note respStmt title

header: edition extent

textcrit: witDetail

Declaration

```
element monogr
{
   att.global.attributes,
   (
      (
         (
            ( author | editor | respStmt ),
            ( author | editor | respStmt )*,
            title+,
            ( editor | respStmt )*
         )
       | ( title+, ( author | editor | respStmt )* )
      )?,
      ( model.noteLike | meeting )*,
      ( edition, ( editor | respStmt )* )*,
      imprint,
      ( imprint | extent | biblScope )*
   )
}
```

Example

```
<biblStruct>
  <analytic>
    <author>Chesnutt, David</author>
    <title>Historical Editions in the States</title>
  </analytic>
  <monogr>
    <title  level="j">Computers and the Humanities</title>
    <imprint>
      <biblScope>25.6</biblScope>
      <date  when="1991-12">(December, 1991):</date>
      <biblScope>377–380</biblScope>
    </imprint>
  </monogr>
</biblStruct>
```

Note May contain specialized bibliographic elements, in a prescribed order.The <monogr> element may only occur only within a <biblStruct>, where its use is mandatory for the description of an monographic level bibliographic item.

<mood> contains information about the grammatical mood of verbs (e.g. indicative, subjunctive, imperative).

Module dictionaries — *9. Dictionaries*

Attributes att.lexicographic (*@expand*, *@norm*, *@split*, *@value*, *@orig*, *@location*, *@mergedIn*, *@opt*)

Used by model.morphLike model.entryPart

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address bibl biblStruct binaryObject cb choice cit corr date del desc distinct email emph expan foreign gap gloss graphic hi index label lb list listBibl measure measureGrp mentioned milestone name note num orig pb ptr q quote ref reg rs said sic soCalled stage term time title unclear

dictionaries: lang oRef oVar pRef pVar

drama: camera caption castList move sound tech view

figures: figure formula table

gaiji: g

header: biblFull

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material msDesc origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName listNym listOrg listPerson listPlace nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att classSpec code eg egXML elementSpec gi ident listRef macroSpec moduleRef moduleSpec specDesc specGrp specGrpRef specList tag val

textcrit: app listWit witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element mood
{
   att.global.attributes,
   att.lexicographic.attributes,
   macro.paraContent}
```

Example Taken from *Wörterbuch der Deutschen Sprache. Veranstaltet und herausgegeben von Joachim Heinrich Campe. Vierter Theil. S - bis - T.* (Braunschweig 1810. In der Schulbuchhandlung):

> Treffen, v. unregelm. ... du triffst, ...

```
<entry>
  <form type="inflected">
    <gramGrp>
```

```
      <per value="2"/>
      <number value="singular"/>
      <tns value="present"/>
      <mood value="indicative"/>
    </gramGrp>
    <form type="personalpronoun">
      <orth>du</orth>
    </form>
    <form type="headword">
      <orth>
        <oVar>triffst</oVar>
      </orth>
    </form>
  </form>
</entry>
```

Note This element is synonymous with **<gram type=mood>**.

<move/> (movement) marks the actual entrance or exit of one or more characters on stage.

Module drama — *7. Performance Texts*

Attributes att.ascribed (*@who*)

@type characterizes the movement, for example as an entrance or exit.

Status Optional

Datatype `data.enumerated`

Suggested values include: **entrance** character is entering the stage.

exit character is exiting the stage.

onStage character moves on stage

@where specifies the direction of a stage movement.

Status Optional

Datatype 1–∞ occurrences of `data.enumerated` separated by whitespace

Sample values include: **L** (left) stage left

R (right) stage right

C (center) centre stage

Note Full blocking information will normally require combinations of values, (for example 'UL' for 'upper stage left') and may also require more detailed encoding of speed, direction etc. Full documentation of any coding system used should be provided in the header.

@perf (performance) identifies the performance or performances in which this movement occurred as specified.

Status Optional

Datatype 1–∞ occurrences of `data.pointer` separated by whitespace

Values The references are derived from the **xml:id** attribute on a <performance> element.

Used by model.stageLike

May contain Empty element

Declaration

```
element move
{
   att.global.attributes,
   att.ascribed.attributes,
   attribute type { "entrance" | "exit" | "onStage" | xsd:Name }?,
   attribute where { list { data.enumerated+ } }?,
   attribute perf { list { data.pointer+ } }?,
   empty
}
```

Example

```
<performance xml:id="perf1">
  <p>First performance</p>
  <castList>
    <castItem>
      <role xml:id="bellaf">Bellafront</role>
    </castItem>
<!-- ... -->
  </castList>
</performance>
<!-- ... -->
<stage type="entrance">
  <move
     who="#bellaf"
     type="enter"
     where="L"
     perf="#perf1"/>
Enter Bellafront mad.
</stage>
```

<msContents>

<msContents> (manuscript contents) describes the intellectual content of a manuscript or manuscript part, either as a series of paragraphs or as a series of structured manuscript items.

Module msdescription — *10. Manuscript Description*

Attributes att.msExcerpt (*@defective*)

@class identifies the text types or classifications applicable to this object.

Status Optional

Datatype `data.code`

Values One or more codes, each of which is used as the identifier for a text classification element supplied in the TEI Header <textClass> element.

Used by msDesc msPart

May contain

core: p

linking: ab

msdescription: msItem msItemStruct summary textLang

Declaration

```
element msContents
{
   att.global.attributes,
```

```
   att.msExcerpt.attributes,
   attribute class { data.code }?,
   ( model.pLike+ | ( summary?, textLang?, ( msItem | msItemStruct )+ ) )
}
```

Example

```
<msContents>
  <p>A collection of Lollard sermons</p>
</msContents>
```

Example

```
<msContents>
  <msItem  n="1">
    <locus>fols. 5r-7v</locus>
    <title>An ABC</title>
    <bibl>
      <title>IMEV</title>
      <biblScope>239</biblScope>
    </bibl>
  </msItem>
  <msItem  n="2">
    <locus>fols. 7v-8v</locus>
    <title  xml:lang="FR">Lenvoy de Chaucer a Scogan</title>
    <bibl>
      <title>IMEV</title>
      <biblScope>3747</biblScope>
    </bibl>
  </msItem>
  <msItem  n="3">
    <locus>fol. 8v</locus>
    <title>Truth</title>
    <bibl>
      <title>IMEV</title>
      <biblScope>809</biblScope>
    </bibl>
  </msItem>
  <msItem  n="4">
    <locus>fols. 8v-10v</locus>
    <title>Birds Praise of Love</title>
    <bibl>
      <title>IMEV</title>
      <biblScope>1506</biblScope>
    </bibl>
  </msItem>
  <msItem  n="5">
    <locus>fols. 10v-11v</locus>
    <title  xml:lang="LA">De amico ad amicam</title>
    <title  xml:lang="LA">Responcio</title>
    <bibl>
      <title>IMEV</title>
      <biblScope>16 & 19</biblScope>
    </bibl>
  </msItem>
  <msItem  n="6">
    <locus>fols. 14r-126v</locus>
    <title>Troilus and Criseyde</title>
```

```
    <note>Bk. 1:71-Bk. 5:1701, with additional losses due to
        mutilation throughout</note>
  </msItem>
</msContents>
```

<msDesc> (manuscript description) contains a description of a single identifiable manuscript.

Module msdescription — *10. Manuscript Description*

Attributes Global attributes only

Used by model.biblLike

May contain

core: head p

linking: ab

msdescription: additional history msContents msIdentifier msPart physDesc

Declaration

```
element msDesc
{
   att.global.attributes,
   (
      msIdentifier,
      model.headLike*,
      (
         model.pLike+
       | ( msContents?, physDesc?, history?, additional?, msPart* )
      )
   )
}
```

Example

```
<msDesc>
  <msIdentifier>
    <settlement>Oxford</settlement>
    <repository>Bodleian Library</repository>
    <idno  type="Bod">MS Poet. Rawl. D. 169.</idno>
  </msIdentifier>
  <msContents>
    <msItem>
      <author>Geoffrey Chaucer</author>
      <title>The Canterbury Tales</title>
    </msItem>
  </msContents>
  <physDesc>
    <objectDesc>
      <p>A parchment codex of 136 folios, measuring approx
            28 by 19 inches, and containing 24 quires.</p>
      <p>The pages are margined and ruled throughout.</p>
      <p>Four hands have been identified in the manuscript: the first 44
            folios being written in two cursive anglicana scripts, while the
            remainder is for the most part in a mixed secretary hand.</p>
    </objectDesc>
```

```
  </physDesc>
</msDesc>
```

<msIdentifier> (manuscript identifier) contains the information required to identify the manuscript being described.

Module msdescription — *10. Manuscript Description*

Attributes Global attributes only

Used by msDesc model.biblPart

May contain

header: idno

msdescription: altIdentifier collection institution msName repository

Declaration

```
element msIdentifier
{
   att.global.attributes,
   (
      (
         (
            model.placeNamePart_sequenceOptional,
            institution?,
            repository,
            collection?,
            idno?
         )
       | msName      ),
      ( altIdentifier | msName )*
   )
}
```

Example

```
<msIdentifier>
  <settlement>San Marino</settlement>
  <repository>Huntington Library</repository>
  <idno>MS.El.26.C.9</idno>
</msIdentifier>
```

<msItem> (manuscript item) describes an individual work or item within the intellectual content of a manuscript or manuscript part.

Module msdescription — *10. Manuscript Description*

Attributes att.msExcerpt (*@defective*)

@class identifies the text types or classifications applicable to this item

Status Optional

Datatype `data.code`

Values One or more codes, each of which is used as the identifier for a text classification element supplied in the TEI Header <textClass> element.

Used by msContents model.msItemPart

May contain

analysis: interp interpGrp span spanGrp

certainty: certainty respons

core: author bibl cb cit editor gap index lb listBibl milestone note p pb quote respStmt title

iso-fs: fLib fs fvLib

linking: ab alt altGrp anchor join joinGrp link linkGrp timeline

msdescription: colophon decoNote explicit filiation finalRubric incipit locus msItem msItemStruct rubric textLang

spoken: incident kinesic pause shift vocal writing

textcrit: witDetail

transcr: addSpan damageSpan delSpan fw space

Declaration

```
element msItem
{
   att.global.attributes,
   att.msExcerpt.attributes,
   attribute class { data.code }?,
   ( locus?, ( model.pLike+ | ( model.msItemPart | model.global )+ ) )
}
```

Example

```
<msItem>
  <locus>ff. 1r-24v</locus>
  <title>Agrip af Noregs konunga sögum</title>
  <incipit>regi oc h<expan>ann</expan> setiho
  <gap  reason="illegible"  extent="7"/>sc
     heim se<expan>m</expan> þio</incipit>
  <explicit>h<expan>on</expan> hev<expan>er</expan>
    <expan>oc</expan>þa buit hesta .ij. aNan viþ
     fé enh<expan>on</expan>o<expan>m</expan> aNan til
     reiþ<expan>ar</expan>
  </explicit>
  <textLang  mainLang="ONI">Old Norse/Icelandic</textLang>
</msItem>
```

<msItemStruct> (structured manuscript item) contains a structured description for an individual work or item within the intellectual content of a manuscript or manuscript part.

Module msdescription — *10. Manuscript Description*

Attributes att.msExcerpt (*@defective*)

@class identifies the text types or classifications applicable to this item

Status Optional

Datatype `data.code`

Values One or more codes, each of which is used as the identifier for a text classification element supplied in the TEI Header <textClass> element.

Used by msContents msItemStruct model.msItemPart

May contain

core: author bibl listBibl note p respStmt title

linking: ab

msdescription: colophon decoNote explicit filiation finalRubric incipit locus msItemStruct rubric textLang

textcrit: witDetail

Declaration

```
element msItemStruct
{
   att.global.attributes,
   att.msExcerpt.attributes,
   attribute class { data.code }?,
   (
      locus?,
      (
         model.pLike+
       | (
            author*,
            respStmt*,
            title*,
            rubric?,
            incipit?,
            msItemStruct*,
            explicit?,
            finalRubric?,
            colophon*,
            decoNote*,
            listBibl*,
            bibl*,
            filiation*,
            model.noteLike*,
            textLang?
         )
      )
   )
}
```

Example

```
<msItemStruct n="2" defective="false" class="biblComm">
  <locus from="24v" to="97v">24v-97v</locus>
  <author>Apringius de Beja</author>
  <title type="uniform" xml:lang="lat">Tractatus in Apocalypsin</title>
  <rubric>Incipit Trac<supplied reason="omitted">ta</supplied>tus
     in apoka<lb/>lipsin eruditissimi uiri <lb/> Apringi ep<expan>iscop</expan>i
     Pacensis eccl<expan>esi</expan>e</rubric>
  <finalRubric>EXPLIC<expan>IT</expan> EXPO<lb/>SITIO APOCALIPSIS
     QVA<expan>M</expan> EXPOSVIT DOM<lb/>NVS APRINGIUS EP<expan>ISCOPU</expan>S.
     DEO GR<expan>ACI</expan>AS AGO. FI<lb/>NITO LABORE ISTO.</finalRubric>
  <bibl>
    <ref target="http://amiBibl.xml#Apringius1900">Apringius</ref>, ed. Férotin</bibl>
  <textLang mainLang="la">Latin</textLang>
</msItemStruct>
```

<msName> (alternative name) contains any form of unstructured alternative name used for a manuscript, such as an 'ocellus nominum', or nickname.

Module msdescription — *10. Manuscript Description*

Attributes att.typed (*@type*, *@subtype*)

Used by msIdentifier

May contain

gaiji: g

Declaration

```
element msName { att.global.attributes, att.typed.attributes, macro.xtext }
```

Example

```
<msName>The Vercelli Book</msName>
```

<msPart> (manuscript part) contains information about an originally distinct manuscript or part of a manuscript, now forming part of a composite manuscript.

Module msdescription — *10. Manuscript Description*

Attributes Global attributes only

Used by msDesc msPart

May contain

core: head p

linking: ab

msdescription: additional altIdentifier history msContents msPart physDesc

Declaration

```
element msPart
{
   att.global.attributes,
   (
      altIdentifier,
      model.headLike*,
      (
         model.pLike+
       | ( msContents?, physDesc?, history?, additional?, msPart* )
      )
   )
}
```

Example

```
<msDesc>
  <msIdentifier>
    <settlement>Amiens</settlement>
    <repository>BibliothÃ¨que Municipale</repository>
    <idno>MS 3</idno>
    <msName>Maurdramnus Bible</msName>
  </msIdentifier>
<!-- other elements here -->
```

```
  <msPart>
    <altIdentifier>
      <idno>MS 6</idno>
    </altIdentifier>
<!-- other information specific to this part here -->
  </msPart>
<!-- more parts here -->
</msDesc>
```

<musicNotation> contains description of type of musical notation.

Module msdescription — *10. Manuscript Description*

Attributes Global attributes only

Used by model.physDescPart

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address bibl biblStruct binaryObject cb choice cit corr date del desc distinct email emph expan foreign gap gloss graphic hi index l label lb lg list listBibl measure measureGrp mentioned milestone name note num orig p pb ptr q quote ref reg rs said sic soCalled sp stage term time title unclear

dictionaries: lang oRef oVar pRef pVar

drama: camera caption castList move sound tech view

figures: figure formula table

gaiji: g

header: biblFull

iso-fs: fLib fs fvLib

linking: ab alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material msDesc origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName listNym listOrg listPerson listPlace nameLink offset orgName persName placeName region roleName settlement state surname

nets: eTree forest forestGrp graph tree

spoken: incident kinesic pause shift u vocal writing

tagdocs: att classSpec code eg egXML elementSpec gi ident listRef macroSpec moduleRef moduleSpec schemaSpec specDesc specGrp specGrpRef specList tag val

textcrit: app listWit witDetail

textstructure: floatingText

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element musicNotation { att.global.attributes, macro.specialPara }
```

Example

```
<musicNotation>
  <p>Square notation of 4-line red staves.</p>
</musicNotation>
```

Example

```
<musicNotation>Neumes in <term>campo aperto</term> of the St. Gall type.
</musicNotation>
```

<name> (name, proper noun) contains a proper noun or noun phrase.

Module core — *3. Elements Available in All TEI Documents*

Attributes att.naming (*@key*, *@ref*, *@nymRef*)

@type indicates the type of the object which is being named by the phrase.

Status Recommended

Datatype `data.enumerated`

Values Values such as person, place, institution, product, acronym.

Used by model.nameLike.agent

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address binaryObject cb choice corr date del distinct email emph expan foreign gap gloss graphic hi index lb measure measureGrp mentioned milestone name note num orig pb ptr ref reg rs sic soCalled term time title unclear

dictionaries: lang oRef oVar pRef pVar

figures: formula

gaiji: g

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident specDesc specList tag val

textcrit: app witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element name
{
   att.global.attributes,
   att.naming.attributes,
   attribute type { data.enumerated }?,
   macro.phraseSeq}
```

Example

```
<name type="person">Thomas Hoccleve</name>
<name type="place">Villingaholt</name>
<name type="org">Vetus Latina Institut</name>
<name type="person" ref="#HOC001">Occleve</name>
```

Note Proper nouns referring to people, places, and organizations may be tagged instead with <persName>, <placeName>, or <orgName>, when the TEI module for names and dates is included.

<nameLink>

<nameLink> contains a connecting phrase or link used within a name but not regarded as part of it, such as *van der* or *of*.

Module namesdates — *13. Names, Dates, People, and Places*

Attributes att.personal (*@full*, *@sort*) att.naming (*@key*, *@ref*, *@nymRef*) att.typed (*@type*, *@subtype*)

Used by model.persNamePart

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address binaryObject cb choice corr date del distinct email emph expan foreign gap gloss graphic hi index lb measure measureGrp mentioned milestone name note num orig pb ptr ref reg rs sic soCalled term time title unclear

dictionaries: lang oRef oVar pRef pVar

figures: formula

gaiji: g

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident specDesc specList tag val

textcrit: app witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element nameLink
{
   att.global.attributes,
   att.personal.attributes,
   att.naming.attributes,
   att.typed.attributes,
   macro.phraseSeq}
```

Example

```
<persName>
  <forename>Frederick</forename>
  <nameLink>van der</nameLink>
  <surname>Tronck</surname>
</persName>
```

Example

```
<persName>
  <forename>Alfred</forename>
  <nameLink>de</nameLink>
  <surname>Musset</surname>
</persName>
```

<namespace> supplies the formal name of the namespace to which the elements documented by its children belong.

Module header — *2. The TEI Header*

Attributes In addition to global attributes

@name the full formal name of the namespace concerned.

Status **Required**

Datatype `data.namespace`

Used by tagsDecl

May contain

header: tagUsage

Declaration

```
element namespace
{
   att.global.attributes,
   attribute name { data.namespace },
   tagUsage+
}
```

Example

```
<namespace name="http://www.tei-c.org/ns/1.0">
  <tagUsage
     gi="hi"
     occurs="28"
```

```
            withId="2"
            render="#it">Used only to mark English words italicised in the copy text
        </tagUsage>
    </namespace>
```

<nationality> contains an informal description of a person's present or past nationality or citizenship.

Module namesdates — *13. Names, Dates, People, and Places*

Attributes att.datableatt.datable.w3c (*@period*, *@when*, *@notBefore*, *@notAfter*, *@from*, *@to*) att.datable.iso (*@when-iso*, *@notBefore-iso*, *@notAfter-iso*, *@from-iso*, *@to-iso*) att.editLike (*@cert*, *@resp*, *@evidence*, *@source*, *@precision*) att.naming (*@key*, *@ref*, *@nymRef*)

Used by model.persTraitLike

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address binaryObject cb choice corr date del distinct email emph expan foreign gap gloss graphic hi index lb measure measureGrp mentioned milestone name note num orig pb ptr ref reg rs sic soCalled term time title unclear

dictionaries: lang oRef oVar pRef pVar

figures: formula

gaiji: g

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident specDesc specList tag val

textcrit: app witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element nationality
{
   att.global.attributes,
   att.datable.w3c.attributes,
   att.datable.iso.attributes,
   att.editLike.attributes,
   att.naming.attributes,
   macro.phraseSeq}
```

Example

```
<nationality key="US" notBefore="1966"> Obtained US Citizenship in 1966</nationality>
```

<node> encodes a node, a possibly labeled point in a graph.

Module nets — *19. Graphs, Networks, and Trees*

Attributes In addition to global attributes

@value provides the value of a node, which is a feature structure or other analytic element.

Status Optional

Datatype `data.pointer`

Values A valid identifier.

@type provides a type for a node.

Status Optional

Datatype `data.enumerated`

Suggested values include: **initial** initial node in a transition network

final final node in a transition network

@adjTo (adjacent to) gives the identifiers of the nodes which are adjacent to the current node.

Status Required when applicable

Datatype 1–∞ occurrences of `data.pointer` separated by whitespace

Values A list of identifiers.

@adjFrom (adjacent from) gives the identifiers of the nodes which are adjacent from the current node.

Status Required when applicable

Datatype 1–∞ occurrences of `data.pointer` separated by whitespace

Values A list of identifiers.

@adj (adjacent) gives the identifiers of the nodes which are both adjacent to and adjacent from the current node.

Status Required when applicable

Datatype 1–∞ occurrences of `data.pointer` separated by whitespace

Values A list of identifiers.

Note Use this attribute instead of the **adjTo** and **adjFrom** attributes when the graph is undirected and vice versa if the graph is directed.

@inDegree gives the in degree of the node, the number of nodes which are adjacent from the given node.

Status Optional

Datatype `data.count`

Values A non-negative integer.

@outDegree gives the out degree of the node, the number of nodes which are adjacent to the given node.

Status Optional

Datatype `data.count`

Values A non-negative integer.

@degree gives the degree of the node, the number of arcs with which the node is incident.

Status Optional

Datatype `data.count`

Values A non-negative integer.

Note Use this attribute instead of the **inDegree** and **outDegree** attributes when the graph is undirected and vice versa if the graph is directed.

Used by graph

May contain

core: label

Declaration

```
element node
{
   att.global.attributes,
   attribute value { data.pointer }?,
   attribute type { "initial" | "final" | xsd:Name }?,
   attribute adjTo { list { data.pointer+ } }?,
   attribute adjFrom { list { data.pointer+ } }?,
   attribute adj { list { data.pointer+ } }?,
   attribute inDegree { data.count }?,
   attribute outDegree { data.count }?,
   attribute degree { data.count }?,
   ( label, label? )?
}
```

Example

```
<node
   xml:id="t6"
   type="final"
   inDegree="2"
   outDegree="0">
  <label>6</label>
</node>
```

Note Zero, one, or two children <label> elements may be present. The first occurence of <label> provides a label for the arc; the second provides a second label for the arc, and should be used if a transducer is being encoded whose actions are associated with nodes rather than with arcs.

<normalization> indicates the extent of normalization or regularization of the original source carried out in converting it to electronic form.

Module header — *2. The TEI Header*

Attributes att.declarable (*@default*)

@source indicates the authority for any normalization carried out.

Status Optional

Datatype `data.pointer`

Values Points to a bibliographic description or other resource documenting the principles underlying the normalization which has been applied.

@method indicates the method adopted to indicate normalizations within the text.

Status Optional

Legal values are: **silent** normalization made silently **[Default]**
markup normalization represented using markup

Used by model.editorialDeclPart

May contain

core: p

linking: ab

Declaration

```
element normalization
{
   att.global.attributes,
   att.declarable.attributes,
   attribute source { data.pointer }?,
   attribute method { "silent" | "markup" }?,
   model.pLike+
}
```

Example

```
<editorialDecl>
  <normalization method="markup">
    <p>Where both upper- and lower-case i, j, u, v, and vv have
        been normalized, to modern 20th century typographical practice,
        the <gi>choice</gi> element has been used to enclose
    <gi>orig</gi> and <gi>reg</gi> elements giving the original and
        new values respectively. ... </p>
  </normalization>
  <normalization method="silent">
    <p>Spacing between words and following punctuation has been
        regularized to zero spaces; spacing between words has been
        regularized to one space.</p>
  </normalization>
  <normalization source="http://www.dict.sztaki.hu/webster">
    <p>Spelling converted throughout to Modern American usage, based on
        Websters 9th Collegiate dictionary.</p>
  </normalization>
</editorialDecl>
```

<note> contains a note or annotation.

Module core — *3. Elements Available in All TEI Documents*

Attributes att.placement (*@place*)

@type describes the type of note.

Status Optional

Datatype `data.enumerated`

Values Values can be taken from any convenient typology of annotation suitable to the work in hand; e.g. annotation, gloss, citation, digression, preliminary, temporary

@resp (responsible party) indicates who is responsible for the annotation: author, editor, translator, etc.

Status Required when applicable

Datatype `data.pointer`

Values a pointer to one of the identifiers declared in the document header, associated with a person asserted as responsible for some aspect of the text's creation, transcription, editing, encoding, or annotation

Note For specialized types of editorial annotation (e.g. for marking corrections, normalizations, cruxes, etc.), see chapter *12. Critical Apparatus*.

@anchored indicates whether the copy text shows the exact place of reference for the note.

Status Optional

Datatype `data.truthValue`

Note In modern texts, notes are usually anchored by means of explicit footnote or endnote symbols. An explicit indication of the phrase or line annotated may however be used instead (e.g. 'page 218, lines 3–4'). The **anchored** attribute indicates whether any explicit location is given, whether by symbol or by prose cross-reference. The value **true** indicates that such an explicit location is indicated in the copy text; the value **false** indicates that the copy text does not indicate a specific place of attachment for the note. If the specific symbols used in the copy text at the location the note is anchored are to be recorded, use the **n** attribute.

@target indicates the point (or points) of attachment for a note, or the beginning of the span to which the note is attached.

Status Required when applicable

Datatype 1–∞ occurrences of `data.pointer` separated by whitespace

Values reference to the **xml:id**s of element(s) which begin at the location in question (e.g. the **xml:id** of an <anchor> element).

Note If **target** and **targetEnd** are to be used to indicate where notes attach to the text, then elements at the appropriate locations (<anchor> elements if necessary) must be given **xml:id** values to be pointed at.

@targetEnd points to the end of the span to which the note is attached, if the note is not embedded in the text at that point.

Status Required when applicable

Datatype 1–∞ occurrences of `data.pointer` separated by whitespace

Values reference to the **xml:id**(s) of element(s) which *end* at the location(s) in question, or to an empty element at the point in question.

Used by altIdentifier model.noteLike

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address bibl biblStruct binaryObject cb choice cit corr date del desc distinct email emph expan foreign gap gloss graphic hi index l label lb lg list listBibl measure measureGrp mentioned milestone name note num orig p pb ptr q quote ref reg rs said sic soCalled sp stage term time title unclear

dictionaries: lang oRef oVar pRef pVar

drama: camera caption castList move sound tech view

figures: figure formula table

gaiji: g

header: biblFull

iso-fs: fLib fs fvLib

linking: ab alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material msDesc origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName listNym listOrg listPerson listPlace nameLink offset orgName persName placeName region roleName settlement state surname

nets: eTree forest forestGrp graph tree

spoken: incident kinesic pause shift u vocal writing

tagdocs: att classSpec code eg egXML elementSpec gi ident listRef macroSpec moduleRef moduleSpec schemaSpec specDesc specGrp specGrpRef specList tag val

textcrit: app listWit witDetail

textstructure: floatingText

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element note
{
   att.global.attributes,
   att.placement.attributes,
   attribute type { data.enumerated }?,
   attribute resp { data.pointer }?,
   attribute anchored { data.truthValue }?,
   attribute target { list { data.pointer+ } }?,
   attribute targetEnd { list { data.pointer+ } }?,
   macro.specialPara}
```

Example

```
And yet it is not only in the great line of Italian
renaissance art, but even in the painterly
<note type="gloss">
  <term xml:lang="de">Malerisch</term>.
This word has, in the German, two distinct meanings, one objective, a
quality residing in the object, the other subjective, a mode of
apprehension and creation. To avoid confusion, they have been
distinguished in English as <mentioned>picturesque</mentioned> and
<mentioned>painterly</mentioned> respectively. (Tr.)
</note>
style of the Dutch genre painters of the seventeenth century that
drapery has this psychological significance.
```

Note The global **n** attribute may be used to supply the symbol or number used to mark the note's point of attachment in the source text, as in the following example:

```
Mevorakh b. Saadya's mother, the matriarch of the family
during the second half of the eleventh century,
<note n="126" anchored="true"> The alleged mention of Judah Nagid's mother in a letter from
1071 is, in fact, a reference to Judah's children; cf. above,
nn. 111 and 54.
</note>
is well known from Geniza documents published by Jacob Mann.
```

However, if notes are numbered in sequence and their numbering can be reconstructed automatically by processing software, it may well be considered unnecessary to record the note numbers.

<notesStmt> (notes statement) collects together any notes providing information about a text additional to that recorded in other parts of the bibliographic description.

Module header — *2. The TEI Header*

Attributes Global attributes only

Used by biblFull fileDesc

May contain

core: note

textcrit: witDetail

Declaration

```
element notesStmt { att.global.attributes, model.noteLike+ }
```

Example

```
<notesStmt>
  <note>Historical commentary provided by Mark Cohen</note>
  <note>OCR scanning done at University of Toronto</note>
</notesStmt>
```

Note Information of different kinds should not be grouped together into the same note.

<num> (number) contains a number, written in any form.

Module core — *3. Elements Available in All TEI Documents*

Attributes In addition to global attributes

@type indicates the type of numeric value.

Status Optional

Datatype `data.enumerated`

Suggested values include: **cardinal** absolute number, e.g. 21, 21.5

ordinal ordinal number, e.g. 21st

fraction fraction, e.g. one half or three-quarters

percentage a percentage

Note If a different typology is desired, other values can be used for this attribute.

@value supplies the value of the number in standard form.

Status Optional

Datatype `data.numeric`

Values a numeric value.

Note The standard form used is defined by the TEI datatype data.numeric.

Used by model.measureLike

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address binaryObject cb choice corr date del distinct email emph expan foreign gap gloss graphic hi index lb measure measureGrp mentioned milestone name note num orig pb ptr ref reg rs sic soCalled term time title unclear

dictionaries: lang oRef oVar pRef pVar

figures: formula

gaiji: g

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident specDesc specList tag val

textcrit: app witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element num
{
   att.global.attributes,
   attribute type
   {
      "cardinal" | "ordinal" | "fraction" | "percentage" | xsd:Name
   }?,
   attribute value { data.numeric }?,
   macro.phraseSeq}
```

Example

```
<p>I reached <num  type="cardinal"  value="21">twenty-one</num>
on my <num  type="ordinal"  value="21">twenty-first</num> birthday...
light travels at <num  value="10E10">10<hi  rend="sup">10</hi>
  </num> cm per second.</p>
```

Note Detailed analyses of quantities and units of measure in historical documents may also use the feature structure mechanism described in chapter *18. Feature Structures*. The <num> element is intended for use in simple applications.

<number> indicates grammatical number associated with a form, as given in a dictionary.

Module dictionaries — *9. Dictionaries*

Attributes att.lexicographic (*@expand*, *@norm*, *@split*, *@value*, *@orig*, *@location*, *@mergedIn*, *@opt*)

Used by model.entryPart model.morphLike

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address bibl biblStruct binaryObject cb choice cit corr date del desc distinct email emph expan foreign gap gloss graphic hi index label lb list listBibl measure measureGrp mentioned milestone name note num orig pb ptr q quote ref reg rs said sic soCalled stage term time title unclear

dictionaries: lang oRef oVar pRef pVar

drama: camera caption castList move sound tech view

figures: figure formula table

gaiji: g

header: biblFull

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material msDesc origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName listNym listOrg listPerson listPlace nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att classSpec code eg egXML elementSpec gi ident listRef macroSpec moduleRef moduleSpec specDesc specGrp specGrpRef specList tag val

textcrit: app listWit witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element number
{
   att.global.attributes,
   att.lexicographic.attributes,
   macro.paraContent}
```

Example

```
<entry>
  <form>
    <orth>wits</orth>
    <pron>wIts</pron>
  </form>
  <gramGrp>
    <number>pl</number>
    <pos>n</pos>
  </gramGrp>
</entry>
```

Note This element is synonymous with **<gram type=num>**.

<numeric/> (numeric value) represents the value part of a feature-value specification which contains a numeric value or range.

Module iso-fs — *18. Feature Structures*

Attributes In addition to global attributes

@value supplies a lower bound for the numeric value represented, and also (if **max** is not supplied) its upper bound.

Status **Required**

Datatype `data.numeric`

Values A real number or integer.

@max supplies an upper bound for the numeric value represented.

Status Optional

Datatype `data.numeric`

Values A real number or integer.

@trunc specifies whether the value represented should be truncated to give an integer value.

Status Optional

Datatype `data.truthValue`

Used by model.featureVal.single

May contain Empty element

Declaration

```
element numeric
{
   att.global.attributes,
   attribute value { data.numeric },
   attribute max { data.numeric }?,
   attribute trunc { data.truthValue }?,
   empty
}
```

Example

```
<numeric value="42"/>
```

This represents the numeric value 42.

Example

```
<numeric  value="42.45"  max="50"  trunc="true"/>
```

This represents any of the nine possible integer values between 42 and 50 inclusive. If the **trunc** attribute had the value FALSE, this example would represent any of the infinite number of numeric values between 42.45 and 50.0

Note It is an error to supply the **max** attribute in the absence of a value for the **value** attribute.

<nym> (canonical name) contains the definition for a canonical name or namepart of any kind.

Module namesdates — *13. Names, Dates, People, and Places*

Attributes att.typed (*@type*, *@subtype*)

@parts points to constituent nyms

Status Optional

Datatype 1–100 occurrences of `data.pointer` separated by whitespace

Used by listNym nym

May contain

core: p

dictionaries: case colloc def etym form gen gramGrp hom hyph iType lbl mood number orth per pos pron re sense stress subc superEntry syll tns usg xr

linking: ab

namesdates: nym

Declaration

```
element nym
{
   att.global.attributes,
   att.typed.attributes,
   attribute parts { list { data.pointer, data.pointer* } }?,
   ( ( model.entryPart* ), ( model.pLike* ), ( nym* ) )
}
```

Example

```
<nym  xml:id="J452">
  <form>
    <orth  xml:lang="en-US">Ian</orth>
    <orth  xml:lang="en-x-Scots">Iain</orth>
  </form>
</nym>
```

<oRef/> (orthographic-form reference) in a dictionary example, indicates a reference to the orthographic form(s) of the headword.

Module dictionaries — *9. Dictionaries*

Attributes att.ptrLike.form (*@target*) att.lexicographic (*@expand*, *@norm*, *@split*, *@value*, *@orig*, *@location*, *@mergedIn*, *@opt*)

@type indicates the kind of typographic modification made to the headword in the reference.

Status Optional

Datatype `data.enumerated`

Sample values include: **cap** (capital) indicates first letter is given as capital

noHyph (no hyphen) indicates that the headword, though a prefix or suffix, loses its hyphen

Used by oVar model.ptrLike.form

May contain Empty element

Declaration

```
element oRef
{
   att.global.attributes,
   att.ptrLike.form.attributes,
   att.lexicographic.attributes,
   attribute type { data.enumerated }?,
   empty
}
```

Example

```
<entry>
  <form>
    <orth>academy</orth>
  </form>
  <cit  type="example">
    <quote>The Royal <oRef  type="cap"/> of Arts</quote>
  </cit>
</entry>
```

<oVar> (orthographic-variant reference) in a dictionary example, indicates a reference to variant orthographic form(s) of the headword.

Module dictionaries — *9. Dictionaries*

Attributes att.ptrLike.form (*@target*) att.lexicographic (*@expand*, *@norm*, *@split*, *@value*, *@orig*, *@location*, *@mergedIn*, *@opt*)

@type indicates the kind of variant involved.

Status Optional

Datatype `data.enumerated`

Sample values include: **pt** (past tense)

pp (past participle)

prp (present participle)

f (feminine)

pl (plural)

Used by model.ptrLike.form

May contain

dictionaries: oRef

gaiji: g

Declaration

```
element oVar
{
   att.global.attributes,
   att.ptrLike.form.attributes,
   att.lexicographic.attributes,
   attribute type { data.enumerated }?,
   ( text | model.gLike | oRef )*
}
```

Example

```
<entry>
  <form>
    <orth>take</orth>
  </form>
  <cit  type="example">
    <quote>Mr Burton <oVar  type="pt">took</oVar> us for French</quote>
  </cit>
</entry>
```

Note Character data or <oRef>.

<objectDesc> contains a description of the physical components making up the object which is being described.

Module msdescription — *10. Manuscript Description*

Attributes In addition to global attributes

@form a short project-specific name identifying the physical form of the carrier, for example as a codex, roll, fragment, partial leaf, cutting etc.

Status Optional

Datatype `data.enumerated`

Values a short project-defined name

Used by model.physDescPart

May contain

core: p

linking: ab

msdescription: layoutDesc supportDesc

Declaration

```
element objectDesc
{
   att.global.attributes,
   attribute form { data.enumerated }?,
   ( model.pLike+ | ( supportDesc?, layoutDesc? ) )
}
```

Example

```
<objectDesc  form="codex">
  <supportDesc  material="mixed">
```

```
      <p>Early modern
      <material>parchment</material> and
      <material>paper</material>.</p>
    </supportDesc>
    <layoutDesc>
      <layout  ruledLines="25 32"/>
    </layoutDesc>
  </objectDesc>
```

<occupation> contains an informal description of a person's trade, profession or occupation.

Module namesdates — *13. Names, Dates, People, and Places*

Attributes att.datableatt.datable.w3c (*@period*, *@when*, *@notBefore*, *@notAfter*, *@from*, *@to*) att.datable.iso (*@when-iso*, *@notBefore-iso*, *@notAfter-iso*, *@from-iso*, *@to-iso*) att.editLike (*@cert*, *@resp*, *@evidence*, *@source*, *@precision*) att.naming (*@key*, *@ref*, *@nymRef*)

@scheme identifies the classification system or taxonomy in use by supplying the identifier of a <taxonomy> element elsewhere in the header.

Status Optional

Datatype `data.pointer`

Values must identify a <taxonomy> element

@code identifies an occupation code defined within the classification system or taxonomy defined by the **scheme** attribute.

Status Optional

Datatype `data.pointer`

Values Must identify a <category> element

Used by model.persStateLike

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address binaryObject cb choice corr date del distinct email emph expan foreign gap gloss graphic hi index lb measure measureGrp mentioned milestone name note num orig pb ptr ref reg rs sic soCalled term time title unclear

dictionaries: lang oRef oVar pRef pVar

figures: formula

gaiji: g

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident specDesc specList tag val

textcrit: app witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element occupation
{
   att.global.attributes,
   att.datable.w3c.attributes,
   att.datable.iso.attributes,
   att.editLike.attributes,
   att.naming.attributes,
   attribute scheme { data.pointer }?,
   attribute code { data.pointer }?,
   macro.phraseSeq}
```

Example

```
<occupation>accountant</occupation>
```

Example

```
<occupation  scheme="#rg"  code="#acc">accountant</occupation>
```

Example

```
<occupation  scheme="#rg"  code="#acc">accountant
with specialist knowledge of oil industry
</occupation>
```

Note The content of this element may be used as an alternative to the more formal specification made possible by its attributes; it may also be used to supplement the formal specification with commentary or clarification.

<offset> that part of a relative temporal or spatial expression which indicates the direction of the offset between the two place names, dates, or times involved in the expression.

Module namesdates — *13. Names, Dates, People, and Places*

Attributes att.typed (*@type*, *@subtype*)

Used by model.offsetLike

May contain

gaiji: g

Declaration

```
element offset { att.global.attributes, att.typed.attributes, macro.xtext }
```

Example

```
<placeName  key="NRPA1">
  <offset>50 metres below the summit of</offset>
```

```
      <geogName>
        <geogFeat>Mount</geogFeat>
        <name>Sinai</name>
      </geogName>
    </placeName>
```

<opener> groups together dateline, byline, salutation, and similar phrases appearing as a preliminary group at the start of a division, especially of a letter.

Module textstructure — *4. Default Text Structure*

Attributes Global attributes only

Used by model.divTopPart

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address binaryObject cb choice corr date del distinct email emph expan foreign gap gloss graphic hi index lb measure measureGrp mentioned milestone name note num orig pb ptr ref reg rs sic soCalled term time title unclear

dictionaries: lang oRef oVar pRef pVar

figures: formula

gaiji: g

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident specDesc specList tag val

textcrit: app witDetail

textstructure: argument byline dateline epigraph salute signed

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element opener
{
   att.global.attributes,
   (
      text
    | model.gLike    | model.phrase    | argument    | byline    | dateline    | epigraph    | salute    | signe
}
```

Example

```
<opener>
  <dateline>Walden, this 29.  of August 1592</dateline>
</opener>
```

Example

```
<opener>
  <dateline>
    <name  type="place">Great Marlborough Street</name>
    <date>November 11, 1848</date>
  </dateline>
  <salute>My dear Sir,</salute>
</opener>
<p>I am sorry to say that absence from town and other
circumstances have prevented me from earlier enquiring...</p>
```

Source: [197]

<org> (organization) provides information about an identifiable organization such as a business, a tribe, or any other grouping of people.

Module namesdates — *13. Names, Dates, People, and Places*

Attributes att.typed (*@type*, *@subtype*) att.editLike (*@cert*, *@resp*, *@evidence*, *@source*, *@precision*)

@role specifies a primary role or classification for the organization.

Status Optional

Datatype 1–∞ occurrences of `data.word` separated by whitespace

Values one or more keywords separated by spaces

Used by listOrg model.personLike

May contain

core: bibl biblStruct desc head label name note p rs

dictionaries: lang

header: biblFull

linking: ab

msdescription: msDesc

namesdates: addName bloc country district forename genName geogFeat geogName nameLink offset org orgName persName person personGrp place placeName region roleName settlement state surname

textcrit: witDetail

Declaration

```
element org
{
   att.global.attributes,
   att.typed.attributes,
   att.editLike.attributes,
   attribute role { list { data.word+ } }?,
   (
      model.headLike*,
      (
         ( model.pLike* )
```

```
       | ( model.labelLike | model.nameLike | model.placeLike )*
      ),
      ( model.noteLike | model.biblLike )*,
      model.personLike*
   )
}
```

Example

```
<org xml:id="JAMs">
  <orgName>Justified Ancients of Mummu</orgName>
  <desc>An underground anarchist collective spearheaded by <persName>Hagbard Celine</persName>,
who fight the Illuminati from a golden submarine, the <name>Leif Ericson</name>
  </desc>
  <bibl>
    <author>Robert Shea</author>
    <author>Robert Anton Wilson</author>
    <title>The Illuminatus! Trilogy</title>
  </bibl>
</org>
```

Note May contain either a prose description organized as paragraphs, or a sequence of more specific demographic elements drawn from the model.personPart class.

<orgName> (organization name) contains an organizational name.

Module namesdates — *13. Names, Dates, People, and Places*

Attributes att.datableatt.datable.w3c (*@period*, *@when*, *@notBefore*, *@notAfter*, *@from*, *@to*) att.datable.iso (*@when-iso*, *@notBefore-iso*, *@notAfter-iso*, *@from-iso*, *@to-iso*) att.editLike (*@cert*, *@resp*, *@evidence*, *@source*, *@precision*) att.personal (*@full*, *@sort*) att.naming (*@key*, *@ref*, *@nymRef*) att.typed (*@type*, *@subtype*)

Used by model.nameLike.agent

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address binaryObject cb choice corr date del distinct email emph expan foreign gap gloss graphic hi index lb measure measureGrp mentioned milestone name note num orig pb ptr ref reg rs sic soCalled term time title unclear

dictionaries: lang oRef oVar pRef pVar

figures: formula

gaiji: g

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident specDesc specList tag val

textcrit: app witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element orgName
{
   att.global.attributes,
   att.datable.w3c.attributes,
   att.datable.iso.attributes,
   att.editLike.attributes,
   att.personal.attributes,
   att.naming.attributes,
   att.typed.attributes,
   macro.phraseSeq}
```

Example

```
About a year back, a question of considerable
interest was agitated in the
<orgName  key="PAS1"  type="voluntary">
  <placeName  key="PEN">Pennsyla.</placeName> Abolition Society
</orgName>....
```

<orig>

(original form) contains a reading which is marked as following the original, rather than being normalized or corrected.

Module core — *3. Elements Available in All TEI Documents*

Attributes Global attributes only

Used by model.pPart.transcriptional model.choicePart

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address bibl biblStruct binaryObject cb choice cit corr date del desc distinct email emph expan foreign gap gloss graphic hi index label lb list listBibl measure measureGrp mentioned milestone name note num orig pb ptr q quote ref reg rs said sic soCalled stage term time title unclear

dictionaries: lang oRef oVar pRef pVar

drama: camera caption castList move sound tech view

figures: figure formula table

gaiji: g

header: biblFull

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material msDesc origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName listNym listOrg listPerson listPlace nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att classSpec code eg egXML elementSpec gi ident listRef macroSpec moduleRef moduleSpec specDesc specGrp specGrpRef specList tag val

textcrit: app listWit witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element orig { att.global.attributes, macro.paraContent }
```

Example If all that is desired is to call attention to the original version in the copy text, <orig> may be used alone:

```
<l>But this
will be a <orig>meere</orig> confusion</l>
<l>And hardly shall we all be <orig>vnderstoode</orig>
</l>
```

Source: [113]

Example More usually, an <orig> will be combined with a regularized form within a <choice> element:

```
<l>But this
will be a <choice>
   <orig>meere</orig>
   <reg>mere</reg>
  </choice> confusion</l>
<l>And hardly shall we all be <choice>
   <orig>vnderstoode</orig>
   <reg>understood</reg>
  </choice>
</l>
```

<origDate>

(origin date) contains any form of date, used to identify the date of origin for a manuscript or manuscript part.

Module msdescription — *10. Manuscript Description*

Attributes att.datableatt.datable.w3c (*@period*, *@when*, *@notBefore*, *@notAfter*, *@from*, *@to*) att.datable.iso (*@when-iso*, *@notBefore-iso*, *@notAfter-iso*, *@from-iso*, *@to-iso*) att.editLike (*@cert*, *@resp*, *@evidence*, *@source*, *@precision*)

Used by model.pPart.msdesc

May contain Character data only

Declaration

```
element origDate
{
   att.global.attributes,
```

```
   att.datable.w3c.attributes,
   att.datable.iso.attributes,
   att.editLike.attributes,
   text
}
```

Example

```
<origDate notBefore="-0300" notAfter="-0200">3rd century BCE</origDate>
```

<origPlace> (origin place) contains any form of place name, used to identify the place of origin for a manuscript or manuscript part.

Module msdescription — *10. Manuscript Description*

Attributes att.editLike (*@cert*, *@resp*, *@evidence*, *@source*, *@precision*)

Used by model.pPart.msdesc

May contain

gaiji: g

Declaration

```
element origPlace
{
   att.global.attributes,
   att.editLike.attributes,
   macro.xtext}
```

Example

```
<origPlace>Birmingham</origPlace>
```

<origin> contains any descriptive or other information concerning the origin of a manuscript or manuscript part.

Module msdescription — *10. Manuscript Description*

Attributes att.editLike (*@cert*, *@resp*, *@evidence*, *@source*, *@precision*) att.datableatt.datable.w3c (*@period*, *@when*, *@notBefore*, *@notAfter*, *@from*, *@to*) att.datable.iso (*@when-iso*, *@notBefore-iso*, *@notAfter-iso*, *@from-iso*, *@to-iso*)

Used by history

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address bibl biblStruct binaryObject cb choice cit corr date del desc distinct email emph expan foreign gap gloss graphic hi index l label lb lg list listBibl measure measureGrp mentioned milestone name note num orig p pb ptr q quote ref reg rs said sic soCalled sp stage term time title unclear

dictionaries: lang oRef oVar pRef pVar

drama: camera caption castList move sound tech view

figures: figure formula table

gaiji: g

header: biblFull

iso-fs: fLib fs fvLib

linking: ab alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material msDesc origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName listNym listOrg listPerson listPlace nameLink offset orgName persName placeName region roleName settlement state surname

nets: eTree forest forestGrp graph tree

spoken: incident kinesic pause shift u vocal writing

tagdocs: att classSpec code eg egXML elementSpec gi ident listRef macroSpec moduleRef moduleSpec schemaSpec specDesc specGrp specGrpRef specList tag val

textcrit: app listWit witDetail

textstructure: floatingText

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element origin
{
   att.global.attributes,
   att.editLike.attributes,
   att.datable.w3c.attributes,
   att.datable.iso.attributes,
   macro.specialPara}
```

Example

```
<origin
   notBefore="1802"
   notAfter="1845"
   evidence="internal"
   resp="#AMH">Copied in <name  type="origPlace">Derby</name>, probably from an
old Flemish original, between 1802 and 1845, according to <persName  xml:id="AMH">Anne-Mette
Hansen</persName>.
</origin>
```

<orth> (orthographic form) gives the orthographic form of a dictionary headword.

Module dictionaries — *9. Dictionaries*

Attributes att.lexicographic (*@expand*, *@norm*, *@split*, *@value*, *@orig*, *@location*, *@mergedIn*, *@opt*)

@type gives the type of spelling.

Status Optional

Datatype `data.enumerated`

Values Any convenient word or phrase, e.g. *lat* (latinate),*std* (standard), *trans* (transliterated), etc.

@extent gives the extent of the orthographic information provided.

Status Optional

Datatype `data.enumerated`

Sample values include: **full** (full form) **[Default]**

pref (prefix)

suff (suffix)

part (partial)

Used by model.entryPart model.formPart

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address bibl biblStruct binaryObject cb choice cit corr date del desc distinct email emph expan foreign gap gloss graphic hi index label lb list listBibl measure measureGrp mentioned milestone name note num orig pb ptr q quote ref reg rs said sic soCalled stage term time title unclear

dictionaries: lang oRef oVar pRef pVar

drama: camera caption castList move sound tech view

figures: figure formula table

gaiji: g

header: biblFull

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material msDesc origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName listNym listOrg listPerson listPlace nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att classSpec code eg egXML elementSpec gi ident listRef macroSpec moduleRef moduleSpec specDesc specGrp specGrpRef specList tag val

textcrit: app listWit witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element orth
{
   att.global.attributes,
   att.lexicographic.attributes,
   attribute type { data.enumerated }?,
   attribute extent { data.enumerated }?,
   macro.paraContent}
```

Example

```
<form type="infl">
  <orth>brags</orth>
  <orth>bragging</orth>
  <orth>bragged</orth>
</form>
```

<p> (paragraph) marks paragraphs in prose.

Module core — *3. Elements Available in All TEI Documents*

Attributes Global attributes only

Used by model.pLike

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address bibl biblStruct binaryObject cb choice cit corr date del desc distinct email emph expan foreign gap gloss graphic hi index label lb list listBibl measure measureGrp mentioned milestone name note num orig pb ptr q quote ref reg rs said sic soCalled stage term time title unclear

dictionaries: lang oRef oVar pRef pVar

drama: camera caption castList move sound tech view

figures: figure formula table

gaiji: g

header: biblFull

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material msDesc origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName listNym listOrg listPerson listPlace nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att classSpec code eg egXML elementSpec gi ident listRef macroSpec moduleRef moduleSpec specDesc specGrp specGrpRef specList tag val

textcrit: app listWit witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element p { att.global.attributes, macro.paraContent }
```

Example

```
<p>Hallgerd was outside.
<q>There is blood on your axe,</q> she said.
<q>What have you done?</q>
</p>
<p>
  <q>I have now arranged that you can be
    married a second time,</q> replied Thjostolf.
</p>
<p>
  <q>Then you must mean that Thorvald is
    dead,</q> she said.
</p>
<p>
  <q>Yes,</q> said Thjostolf.
<q>And now you must think up some plan for me.</q>
</p>
```

Source: [148]

<pRef/> (pronunciation reference) in a dictionary example, indicates a reference to the pronunciation(s) of the headword.

Module dictionaries — *9. Dictionaries*

Attributes att.ptrLike.form (*@target*) att.lexicographic (*@expand*, *@norm*, *@split*, *@value*, *@orig*, *@location*, *@mergedIn*, *@opt*)

Used by pVar model.ptrLike.form

May contain Empty element

Declaration

```
element pRef
{
   att.global.attributes,
   att.ptrLike.form.attributes,
   att.lexicographic.attributes,
   empty
}
```

<pVar> (pronunciation-variant reference) in a dictionary example, indicates a reference to variant pronunciation(s) of the headword.

Module dictionaries — *9. Dictionaries*

Attributes att.ptrLike.form (*@target*) att.lexicographic (*@expand*, *@norm*, *@split*, *@value*, *@orig*, *@location*, *@mergedIn*, *@opt*)

Used by model.ptrLike.form

May contain

dictionaries: pRef

gaiji: g

Declaration

```
element pVar
{
   att.global.attributes,
   att.ptrLike.form.attributes,
   att.lexicographic.attributes,
   ( text | model.gLike | pRef )*
}
```

Note Character data or <pRef>.

<particDesc> (participation description) describes the identifiable speakers, voices, or other participants in a linguistic interaction.

Module corpus — *15. Language Corpora*

Attributes att.declarable (*@default*)

Used by model.profileDescPart

May contain

core: p

linking: ab

namesdates: listPerson org person personGrp

Declaration

```
element particDesc
{
   att.global.attributes,
   att.declarable.attributes,
   ( model.pLike+ | ( model.personLike | listPerson )+ )
}
```

Example

```
<particDesc>
  <listPerson>
    <person xml:id="P-1234" sex="2" age="mid">
      <p>Female informant, well-educated, born in Shropshire
            UK, 12 Jan 1950, of unknown occupation.
            Speaks French fluently. Socio-Economic status B2.</p>
    </person>
    <person xml:id="P-4332" sex="1">
      <persName>
        <surname>Hancock</surname>
        <forename>Antony</forename>
        <forename>Aloysius</forename>
        <forename>St John</forename>
      </persName>
      <residence notAfter="1959">
        <address>
          <street>Railway Cuttings</street>
          <settlement>East Cheam</settlement>
        </address>
      </residence>
      <occupation>comedian</occupation>
```

```
    </person>
    <relationGrp>
      <relation  type="personal"  name="spouse"  mutual="#P-1234 #P-4332"/>
    </relationGrp>
  </listPerson>
</particDesc>
```

This example shows both a very simple person description, and a very detailed one, using some of the more specialised elements from the module for Names and Dates.

Note May contain a prose description organized as paragraphs, or a structured list of persons and person groups, with an optional formal specification of any relationships amongst them.

<pause/> a pause either between or within utterances.

Module spoken — *8. Transcriptions of Speech*

Attributes att.timed (*@start*, *@end*) att.duration.w3c (*@dur*) att.typed (*@type*, *@subtype*) att.ascribed (*@who*)

Used by model.global.spoken

May contain Empty element

Declaration

```
element pause
{
   att.global.attributes,
   att.timed.attributes,
   att.duration.w3c.attributes,
   att.typed.attributes,
   att.ascribed.attributes,
   empty
}
```

Example

```
<pause  dur="PT42S"  type="pregnant"/>
```

<pb/> (page break) marks the boundary between one page of a text and the next in a standard reference system.

Module core — *3. Elements Available in All TEI Documents*

Attributes In addition to global attributes

@ed (edition) indicates the edition or version in which the page break is located at this point

Status Recommended

Datatype `data.code`

Values Any string of characters; usually a siglum conventionally used for the edition.

Used by model.milestoneLike

May contain Empty element

Declaration

```
element pb { att.global.attributes, attribute ed { data.code }?, empty }
```

Example Page numbers may vary in different editions of a text.

```
<p>...
<pb  n="145"  ed="ed2"/>
<!-- Page 145 in edition "ed2" starts here -->
...
<pb  n="283"  ed="ed1"/>
<!-- Page 283 in edition "ed1" starts here-->
...
</p>
```

Example A page break may be associated with a facsimile image of the page it introduces by means of the **facs** attribute

```
<TEI>
  <teiHeader>
<!--...-->
  </teiHeader>
  <text>
    <pb  n="1"  facs="page1.png"/>
<!-- page1.png contains an image of the page; the text it contains is encoded here -->
    <pb  n="2"  facs="page2.png"/>
<!-- similarly, for page 2 -->
  </text>
</TEI>
```

Note On this element, the global **n** attribute indicates the number or other value associated with the page which follows the point of insertion of this <pb>. Encoders should adopt a clear and consistent policy as to whether the numbers associated with page breaks relate to the physical sequence number of the page or the page number or signature printed on it. By convention, <pb> elements should appear at the start of the page to which they refer.

<per> (person) contains an indication of the grammatical person (1st, 2nd, 3rd, etc.) associated with a given inflected form in a dictionary.

Module dictionaries — *9. Dictionaries*

Attributes att.lexicographic (*@expand*, *@norm*, *@split*, *@value*, *@orig*, *@location*, *@mergedIn*, *@opt*)

Used by model.morphLike model.entryPart

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address bibl biblStruct binaryObject cb choice cit corr date del desc distinct email emph expan foreign gap gloss graphic hi index label lb list listBibl measure measureGrp mentioned milestone name note num orig pb ptr q quote ref reg rs said sic soCalled stage term time title unclear

dictionaries: lang oRef oVar pRef pVar

drama: camera caption castList move sound tech view

figures: figure formula table

gaiji: g

header: biblFull

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material msDesc origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName listNym listOrg listPerson listPlace nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att classSpec code eg egXML elementSpec gi ident listRef macroSpec moduleRef moduleSpec specDesc specGrp specGrpRef specList tag val

textcrit: app listWit witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element per
{
   att.global.attributes,
   att.lexicographic.attributes,
   macro.paraContent}
```

Example Taken from *Wörterbuch der Deutschen Sprache. Veranstaltet und herausgegeben von Joachim Heinrich Campe. Vierter Theil. S - bis - T.* (Braunschweig 1810. In der Schulbuchhandlung):

Treffen, v. unregelm. ... du triffst, ...

```
<entry>
  <form type="inflected">
    <gramGrp>
      <per value="2"/>
      <number value="singular"/>
      <tns value="present"/>
      <mood value="indicative"/>
    </gramGrp>
    <form type="personalpronoun">
      <orth>du</orth>
    </form>
    <form type="headword">
      <orth>
        <oVar>triffst</oVar>
      </orth>
    </form>
  </form>
</entry>
```

Note This element is synonymous with **<gram type="person">**.

<performance> contains a section of front or back matter describing how a dramatic piece is to be performed in general or how it was performed on some specific occasion.

Module drama — *7. Performance Texts*

Attributes Global attributes only

Used by model.frontPart.drama

May contain

analysis: interp interpGrp span spanGrp

certainty: certainty respons

core: bibl biblStruct cb cit desc gap head index l label lb lg list listBibl meeting milestone note p pb q quote said sp stage

dictionaries: entry entryFree superEntry

drama: camera caption castList move sound tech view

figures: figure table

header: biblFull

iso-fs: fLib fs fvLib

linking: ab alt altGrp anchor join joinGrp link linkGrp timeline

msdescription: msDesc

namesdates: listNym listOrg listPerson listPlace

nets: eTree forest forestGrp graph tree

spoken: incident kinesic pause shift u vocal writing

tagdocs: classSpec eg egXML elementSpec listRef macroSpec moduleRef moduleSpec schemaSpec specGrp specGrpRef

textcrit: listWit witDetail

textstructure: argument byline closer dateline docAuthor docDate epigraph floatingText opener postscript salute signed trailer

transcr: addSpan damageSpan delSpan fw space

Declaration

```
element performance
{
   att.global.attributes,
   (
      ( model.divTop | model.global )*,
      ( ( model.common ), model.global* )+,
      ( ( model.divBottom ), model.global* )*
   )
}
```

Example

```
<performance>
  <p>
    <rs  type="place">Gateway Theatre, Edinburgh</rs>,
  <date>6 September 1948</date>
    <castList>
      <castItem>
        <role>Anath Bithiah</role>
        <actor>Athene Seyler</actor>
```

```
      </castItem>
      <castItem>
        <role>Shendi</role>
        <actor>Robert Rietty</actor>
      </castItem>
    </castList>
  </p>
  <p>Directed by <name>E. Martin Browne</name>
  </p>
</performance>
```

Example

```
<performance>
  <p>Cast of the original production at the
  <rs  type="place">Savoy Theatre, London,</rs>
     on <date>September 24, 1907</date>
    <castList>
      <castItem>Colonel Hope : Mr A.E.George</castItem>
    </castList>
  </p>
</performance>
```

Note contains paragraphs and an optional cast list only.

<persName>

(personal name) contains a proper noun or proper-noun phrase referring to a person, possibly including any or all of the person's forenames, surnames, honorifics, added names, etc.

Module namesdates — *13. Names, Dates, People, and Places*

Attributes att.datableatt.datable.w3c (*@period*, *@when*, *@notBefore*, *@notAfter*, *@from*, *@to*) att.datable.iso (*@when-iso*, *@notBefore-iso*, *@notAfter-iso*, *@from-iso*, *@to-iso*) att.editLike (*@cert*, *@resp*, *@evidence*, *@source*, *@precision*) att.personal (*@full*, *@sort*) att.naming (*@key*, *@ref*, *@nymRef*) att.typed (*@type*, *@subtype*)

Used by model.persStateLike model.nameLike.agent

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address binaryObject cb choice corr date del distinct email emph expan foreign gap gloss graphic hi index lb measure measureGrp mentioned milestone name note num orig pb ptr ref reg rs sic soCalled term time title unclear

dictionaries: lang oRef oVar pRef pVar

figures: formula

gaiji: g

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat

geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident specDesc specList tag val

textcrit: app witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element persName
{
   att.global.attributes,
   att.datable.w3c.attributes,
   att.datable.iso.attributes,
   att.editLike.attributes,
   att.personal.attributes,
   att.naming.attributes,
   att.typed.attributes,
   macro.phraseSeq}
```

Example

```
<persName>
  <forename>Edward</forename>
  <forename>George</forename>
  <surname  type="linked">Bulwer-Lytton</surname>,
<roleName>Baron Lytton of
  <placeName>Knebworth</placeName>
  </roleName>
</persName>
```

<person> provides information about an identifiable individual, for example a participant in a language interaction, or a person referred to in a historical source.

Module namesdates — *13. Names, Dates, People, and Places*

Attributes att.editLike (*@cert*, *@resp*, *@evidence*, *@source*, *@precision*)

@role specifies a primary role or classification for the person.

Status Optional

Datatype 1–∞ occurrences of `data.enumerated` separated by whitespace

Values the value should be chosen from a set of user-defined and user-documented keywords declared in the customization file

@sex specifies the sex of the person.

Status Optional

Datatype `data.sex`

@age specifies an age group for the person.

Status Optional

Datatype `data.enumerated`

Values the value should be chosen from a set of user-defined and user-documented keywords declared in the customization file; possibilities include **infant**, **child**, **teen**, **adult**, and **senior**.

Used by model.personLike

May contain

analysis: interp interpGrp span spanGrp

certainty: certainty respons

core: bibl cb gap index lb milestone note p pb

iso-fs: fLib fs fvLib

linking: ab alt altGrp anchor join joinGrp link linkGrp timeline

namesdates: affiliation age birth death education event faith floruit langKnowledge nationality occupation persName residence sex socecStatus state trait

spoken: incident kinesic pause shift vocal writing

textcrit: witDetail

transcr: addSpan damageSpan delSpan fw space

Declaration

```
element person
{
   att.global.attributes,
   att.editLike.attributes,
   attribute role { list { data.enumerated+ } }?,
   attribute sex { data.sex }?,
   attribute age { data.enumerated }?,
   ( model.pLike+ | ( model.personPart | model.global )* )
}
```

Example

```
<person sex="2" age="mid">
  <p>Female respondent, well-educated, born in Shropshire
     UK, 12 Jan 1950, of unknown occupation.
     Speaks French fluently. Socio-Economic status B2.</p>
</person>
```

Example

```
<person xml:id="Ovi01" sex="1" role="poet">
  <persName xml:lang="en">Ovid</persName>
  <persName xml:lang="la">Publius Ovidius Naso</persName>
  <birth when="-0044-03-20"> 20 March 43 BC
  <placeName>
     <settlement type="city">Sulmona</settlement>
     <country key="IT">Italy</country>
   </placeName>
  </birth>
  <death notBefore="0017" notAfter="0018">17 or 18 AD
  <placeName>
     <settlement type="city">Tomis (Constanta)</settlement>
     <country key="RO">Romania</country>
   </placeName>
```

```
      </death>
    </person>
```

Note May contain either a prose description organized as paragraphs, or a sequence of more specific demographic elements drawn from the model.personPart class.

<personGrp> (personal group) describes a group of individuals treated as a single person for analytic purposes.

Module namesdates — *13. Names, Dates, People, and Places*

Attributes In addition to global attributes

@role specifies the role of this group of participants in the interaction.
- *Status* Optional
- *Datatype* `data.enumerated`
- *Values* the value should be chosen from a set of user-defined and user-documented keywords declared in the customization file

@sex specifies the sex of the participant group.
- *Status* Optional
- *Datatype* `data.sex | "mixed"`

@age specifies the age group of the participants.
- *Status* Optional
- *Datatype* `data.enumerated`
- *Values* the value should be chosen from a set of user-defined and user-documented keywords declared in the customization file

@size specifies the size or approximate size of the group.
- *Status* Optional
- *Datatype* 1–∞ occurrences of `data.word` separated by whitespace
- *Values* may contain a number and an indication of accuracy, e.g. **approx 200**

Used by model.personLike

May contain

core: bibl p

linking: ab

namesdates: affiliation age birth death education event faith floruit langKnowledge nationality occupation persName residence sex socecStatus state trait

Declaration

```
element personGrp
{
   att.global.attributes,
   attribute role { data.enumerated }?,
   attribute sex { data.sex | "mixed" }?,
   attribute age { data.enumerated }?,
   attribute size { list { data.word+ } }?,
   ( model.pLike+ | model.personPart* )
}
```

Example

```
<personGrp
   xml:id="pg1"
   role="audience"
   sex="mixed"
   size="approx 50"/>
```

Note May contain a prose description organized as paragraphs, or any sequence of demographic elements in any combination.The global **xml:id** attribute should be used to identify each speaking participant in a spoken text if the **who** attribute is specified on individual utterances.

<phr> (phrase) represents a grammatical phrase.

Module analysis — *17. Simple Analytic Mechanisms*

Attributes att.segLike (*@type*, *@function*, *@part*) att.metrical (*@met*, *@real*, *@rhyme*)

Used by model.segLike

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address binaryObject cb choice corr date del distinct email emph expan foreign gap gloss graphic hi index lb measure measureGrp mentioned milestone name note num orig pb ptr ref reg rs sic soCalled term time title unclear

dictionaries: lang oRef oVar pRef pVar

figures: formula

gaiji: g

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident specDesc specList tag val

textcrit: app witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element phr
{
   att.global.attributes,
   att.segLike.attributes,
   att.metrical.attributes,
   macro.phraseSeq}
```

Example

```
<phr type="verb" function="extraposted_modifier">To talk
<phr type="preposition" function="complement">of
  <phr type="noun" function="object">many things</phr>
  </phr>
</phr>
```

<physDesc> (physical description) contains a full physical description of a manuscript or manuscript part, optionally subdivided using more specialised elements from the model.physDescPart class.

Module msdescription — *10. Manuscript Description*

Attributes Global attributes only

Used by msDesc msPart

May contain

core: p

linking: ab

Declaration

```
element physDesc
{
   att.global.attributes,
   ( model.pLike+ | ( model.physDescPart_sequenceOptional ) )
}
```

Example

```
<physDesc>
  <objectDesc form="codex">
    <supportDesc material="perg">
      <support>Parchment.</support>
      <extent>i + 55 leaves
      <dimensions scope="all" type="leaf" unit="inch">
         <height>7Â¼</height>
         <width>5â??</width>
       </dimensions>
      </extent>
    </supportDesc>
    <layoutDesc>
      <layout columns="2">In double columns.</layout>
    </layoutDesc>
  </objectDesc>
  <handDesc>
    <p>Written in more than one hand.</p>
  </handDesc>
  <decoDesc>
    <p>With a few coloured capitals.</p>
  </decoDesc>
</physDesc>
```

<place> contains data about a geographic location

Module namesdates — *13. Names, Dates, People, and Places*

Attributes att.typed (*@type*, *@subtype*) att.editLike (*@cert*, *@resp*, *@evidence*, *@source*, *@precision*)

Used by model.placeLike

May contain

core: bibl biblStruct desc head label note p

header: biblFull

linking: ab

msdescription: msDesc

namesdates: bloc climate country district event geogName listPlace location place placeName population region settlement state terrain trait

textcrit: witDetail

Declaration

```
element place
{
   att.global.attributes,
   att.typed.attributes,
   att.editLike.attributes,
   (
      model.headLike*,
      (
         ( model.pLike* )
       | (
            model.labelLike          | model.placeStateLike          | model.placeTraitLike          | model.pla
      ),
      ( model.noteLike | model.biblLike )*,
      ( model.placeLike | listPlace )*
   )
}
```

Example

```
<place>
  <country>Lithuania</country>
  <country  xml:lang="lt">Lietuva</country>
  <place>
    <settlement>Vilnius</settlement>
  </place>
  <place>
    <settlement>Kaunas</settlement>
  </place>
</place>
```

<placeName> contains an absolute or relative place name.

Module namesdates — *13. Names, Dates, People, and Places*

Attributes att.naming (*@key*, *@ref*, *@nymRef*) att.typed (*@type*, *@subtype*) att.datableatt.datable.w3c (*@period*, *@when*, *@notBefore*, *@notAfter*, *@from*, *@to*) att.datable.iso (*@when-iso*, *@notBefore-iso*, *@notAfter-iso*, *@from-iso*, *@to-iso*) att.editLike (*@cert*, *@resp*, *@evidence*, *@source*, *@precision*)

Used by model.placeNamePart

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address binaryObject cb choice corr date del distinct email emph expan foreign gap gloss graphic hi index lb measure measureGrp mentioned milestone name note num orig pb ptr ref reg rs sic soCalled term time title unclear

dictionaries: lang oRef oVar pRef pVar

figures: formula

gaiji: g

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident specDesc specList tag val

textcrit: app witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element placeName
{
   att.global.attributes,
   att.naming.attributes,
   att.typed.attributes,
   att.datable.w3c.attributes,
   att.datable.iso.attributes,
   att.editLike.attributes,
   macro.phraseSeq}
```

Example

```
<placeName>
  <settlement>Rochester</settlement>
  <region>New York</region>
</placeName>
```

Example

```
<placeName>
  <geogName>Arrochar Alps</geogName>
  <region>Argylshire</region>
</placeName>
```

Example

```
<placeName>
  <measure>10 miles</measure>
  <offset>Northeast of</offset>
  <settlement>Attica</settlement>
</placeName>
```

<population> contains information about the population of a place.

Module namesdates — *13. Names, Dates, People, and Places*

Attributes att.datableatt.datable.w3c (*@period*, *@when*, *@notBefore*, *@notAfter*, *@from*, *@to*) att.datable.iso (*@when-iso*, *@notBefore-iso*, *@notAfter-iso*, *@from-iso*, *@to-iso*) att.editLike (*@cert*, *@resp*, *@evidence*, *@source*, *@precision*) att.naming (*@key*, *@ref*, *@nymRef*) att.typed (*@type*, *@subtype*)

Used by population model.placeTraitLike

May contain

core: bibl biblStruct desc head label note p

header: biblFull

linking: ab

msdescription: msDesc

namesdates: population

textcrit: witDetail

Declaration

```
element population
{
   att.global.attributes,
   att.datable.w3c.attributes,
   att.datable.iso.attributes,
   att.editLike.attributes,
   att.naming.attributes,
   att.typed.attributes,
   (
      model.headLike*,
      (
         ( ( model.pLike+ ) | ( model.labelLike+ ) ),
         ( model.noteLike | model.biblLike )*
      )?,
      population*
   )
}
```

Example

```
<population  when="2001-04"  resp="UKCensus">
  <population  type="white">
    <desc>54153898</desc>
  </population>
  <population  type="asian">
    <desc>11811423</desc>
  </population>
```

```
  <population type="black">
    <desc>1148738</desc>
  </population>
  <population type="mixed">
    <desc>677117</desc>
  </population>
  <population type="chinese">
    <desc>247403</desc>
  </population>
  <population type="other">
    <desc>230615</desc>
  </population>
</population>
```

<pos> (part of speech) indicates the part of speech assigned to a dictionary headword such as noun, verb, or adjective.

Module dictionaries — *9. Dictionaries*

Attributes att.lexicographic (*@expand*, *@norm*, *@split*, *@value*, *@orig*, *@location*, *@mergedIn*, *@opt*)

Used by model.entryPart model.gramPart

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address bibl biblStruct binaryObject cb choice cit corr date del desc distinct email emph expan foreign gap gloss graphic hi index label lb list listBibl measure measureGrp mentioned milestone name note num orig pb ptr q quote ref reg rs said sic soCalled stage term time title unclear

dictionaries: lang oRef oVar pRef pVar

drama: camera caption castList move sound tech view

figures: figure formula table

gaiji: g

header: biblFull

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material msDesc origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName listNym listOrg listPerson listPlace nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att classSpec code eg egXML elementSpec gi ident listRef macroSpec moduleRef moduleSpec specDesc specGrp specGrpRef specList tag val

textcrit: app listWit witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element pos
{
   att.global.attributes,
   att.lexicographic.attributes,
   macro.paraContent}
```

Example

```
<entry>
  <form>
    <orth>isotope</orth>
  </form>
  <gramGrp>
    <pos>adj</pos>
  </gramGrp>
</entry>
```

<postBox> (postal box or post office box) contains a number or other identifier for some postal delivery point other than a street address.

Module core — *3. Elements Available in All TEI Documents*

Attributes Global attributes only

Used by model.addrPart

May contain Character data only

Declaration `element postBox { att.global.attributes, text }`

Example

```
<postBox>P.O. Box 280</postBox>
```

Example

```
<postBox>Postbus 532</postBox>
```

Note The position and nature of postal codes is highly country-specific; the conventions appropriate to the country concerned should be used.

<postCode> (postal code) contains a numerical or alphanumeric code used as part of a postal address to simplify sorting or delivery of mail.

Module core — *3. Elements Available in All TEI Documents*

Attributes Global attributes only

Used by model.addrPart

May contain Character data only

Declaration `element postCode { att.global.attributes, text }`

Example

```
<postCode>HR1 3LR</postCode>
```

Example

```
<postCode>60142-7</postCode>
```

Note The position and nature of postal codes is highly country-specific; the conventions appropriate to the country concerned should be used.

<postscript> contains a postscript, e.g. to a letter.

Module textstructure — *4. Default Text Structure*

Attributes Global attributes only

Used by model.divBottomPart

May contain

analysis: interp interpGrp span spanGrp

certainty: certainty respons

core: bibl biblStruct cb cit desc gap index l label lb lg list listBibl milestone note p pb q quote said sp stage

dictionaries: entry entryFree superEntry

drama: camera caption castList move sound tech view

figures: figure table

header: biblFull

iso-fs: fLib fs fvLib

linking: ab alt altGrp anchor join joinGrp link linkGrp timeline

msdescription: msDesc

namesdates: listNym listOrg listPerson listPlace

nets: eTree forest forestGrp graph tree

spoken: incident kinesic pause shift u vocal writing

tagdocs: classSpec eg egXML elementSpec listRef macroSpec moduleRef moduleSpec schemaSpec specGrp specGrpRef

textcrit: listWit witDetail

textstructure: floatingText

transcr: addSpan damageSpan delSpan fw space

Declaration

```
element postscript
{
   att.global.attributes,
   ( ( model.common ) | ( model.global ) )*
}
```

Example

```
<div type="letter">
  <opener>
```

```
    <dateline>
      <placeName>Rimaone</placeName>
      <date  when="2006-11-21">21 Nov 06</date>
    </dateline>
    <salute>Dear Susan,</salute>
  </opener>
  <p>Thank you very much for the assistance splitting those
     logs. I'm sorry about the misunderstanding as to the size of
     the task. I really was not asking for help, only to borrow the
     axe. Hope you had fun in any case.</p>
  <closer>
    <salute>Sincerely yours,</salute>
    <signed>Seymour</signed>
  </closer>
  <postscript>
    <label>P.S.</label>
    <p>The collision occured on <date  when="2001-07-06">06 Jul 01</date>.</p>
  </postscript>
</div>
```

<preparedness> describes the extent to which a text may be regarded as prepared or spontaneous.

Module corpus — *15. Language Corpora*

Attributes In addition to global attributes

@type a keyword characterizing the type of preparedness.

Status Optional

Datatype `data.enumerated`

Sample values include: **none** spontaneous or unprepared

scripted follows a script

formulaic follows a predefined set of conventions

revised polished or revised before presentation

Used by model.textDescPart

May contain

analysis: interp interpGrp span spanGrp

certainty: certainty respons

core: abbr address cb choice date distinct email emph expan foreign gap gloss index lb measure measureGrp mentioned milestone name note num pb ptr ref rs soCalled term time title

dictionaries: lang

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident tag val

textcrit: witDetail

transcr: addSpan am damageSpan delSpan ex fw handShift space subst

Declaration

```
element preparedness
{
   att.global.attributes,
   attribute type { data.enumerated }?,
   macro.phraseSeq.limited}
```

Example

```
<preparedness  type="none"/>
```

<principal> (principal researcher) supplies the name of the principal researcher responsible for the creation of an electronic text.

Module header — *2. The TEI Header*

Attributes Global attributes only

Used by titleStmt

May contain

analysis: interp interpGrp span spanGrp

certainty: certainty respons

core: abbr address cb choice date distinct email emph expan foreign gap gloss index lb measure measureGrp mentioned milestone name note num pb ptr ref rs soCalled term time title

dictionaries: lang

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident tag val

textcrit: witDetail

transcr: addSpan am damageSpan delSpan ex fw handShift space subst

Declaration

```
element principal { att.global.attributes, macro.phraseSeq.limited }
```

Example

```
<principal>Gary Taylor</principal>
```

<profileDesc>

(text-profile description) provides a detailed description of non-bibliographic aspects of a text, specifically the languages and sublanguages used, the situation in which it was produced, the participants and their setting.

Module header — *2. The TEI Header*

Attributes Global attributes only

Used by model.headerPart

May contain

corpus: particDesc settingDesc textDesc

header: creation langUsage textClass

transcr: handNotes

Declaration

```
element profileDesc
{
   att.global.attributes,
   ( creation?, model.profileDescPart* )
}
```

Example

```
<profileDesc>
  <langUsage>
    <language ident="fr">French</language>
  </langUsage>
  <textDesc n="novel">
    <channel mode="w">print; part issues</channel>
    <constitution type="single"/>
    <derivation type="original"/>
    <domain type="art"/>
    <factuality type="fiction"/>
    <interaction type="none"/>
    <preparedness type="prepared"/>
    <purpose type="entertain" degree="high"/>
    <purpose type="inform" degree="medium"/>
  </textDesc>
  <settingDesc>
    <setting>
      <name>Paris, France</name>
      <time>Late 19th century</time>
    </setting>
  </settingDesc>
</profileDesc>
```

<projectDesc>

(project description) describes in detail the aim or purpose for which an electronic file was encoded, together with any other relevant information concerning the process by which it was assembled or collected.

Module header — *2. The TEI Header*

Attributes att.declarable (*@default*)

Used by model.encodingPart

May contain

core: p

linking: ab

Declaration

```
element projectDesc
{
   att.global.attributes,
   att.declarable.attributes,
   model.pLike+
}
```

Example

```
<projectDesc>
  <p>Texts collected for use in the Claremont Shakespeare
     Clinic, June 1990</p>
</projectDesc>
```

<prologue> contains the prologue to a drama, typically spoken by an actor out of character, possibly in association with a particular performance or venue.

Module drama — *7. Performance Texts*

Attributes Global attributes only

Used by model.frontPart.drama

May contain

analysis: interp interpGrp span spanGrp

certainty: certainty respons

core: bibl biblStruct cb cit desc gap head index l label lb lg list listBibl meeting milestone note p pb q quote said sp stage

dictionaries: entry entryFree superEntry

drama: camera caption castList move sound tech view

figures: figure table

header: biblFull

iso-fs: fLib fs fvLib

linking: ab alt altGrp anchor join joinGrp link linkGrp timeline

msdescription: msDesc

namesdates: listNym listOrg listPerson listPlace

nets: eTree forest forestGrp graph tree

spoken: incident kinesic pause shift u vocal writing

tagdocs: classSpec eg egXML elementSpec listRef macroSpec moduleRef moduleSpec schemaSpec specGrp specGrpRef

textcrit: listWit witDetail

textstructure: argument byline closer dateline docAuthor docDate epigraph floatingText opener postscript salute signed trailer

transcr: addSpan damageSpan delSpan fw space

Declaration

```
element prologue
{
   att.global.attributes,
   (
      ( model.divTop | model.global )*,
      ( ( model.common ), model.global* )+,
      ( ( model.divBottom ), model.global* )*
   )
}
```

Example

```
<prologue>
  <sp>
    <l>Wits, like physicians never can agree,</l>
    <l>When of a different society.</l>
    <l>New plays are stuffed with wits, and with deboches,</l>
    <l>That crowd and sweat like cits in May-Day coaches.</l>
  </sp>
  <trailer>Written by a person of quality</trailer>
</prologue>
```

Source: [13]

<pron> (pronunciation) contains the pronunciation(s) of the word.

Module dictionaries — *9. Dictionaries*

Attributes att.lexicographic (*@expand*, *@norm*, *@split*, *@value*, *@orig*, *@location*, *@mergedIn*, *@opt*)

@extent indicates whether the pronunciation is for whole word or part.

Status Optional

Datatype `data.enumerated`

Sample values include: **full** (full form) **[Default]**

pref (prefix)

suff (suffix)

part (partial)

@notation indicates what notation is used for the pronunciation, if more than one occurs in the machine-readable dictionary.

Status Required when applicable

Datatype `data.enumerated`

Values Sample values: IPA, Murray, ...

Used by model.entryPart model.formPart

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address bibl biblStruct binaryObject cb choice cit corr date del desc distinct email emph expan foreign gap gloss graphic hi index label lb list listBibl measure measureGrp mentioned milestone name note num orig pb ptr q quote ref reg rs said sic soCalled stage term time title unclear

dictionaries: lang oRef oVar pRef pVar

drama: camera caption castList move sound tech view

figures: figure formula table

gaiji: g

header: biblFull

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material msDesc origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName listNym listOrg listPerson listPlace nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att classSpec code eg egXML elementSpec gi ident listRef macroSpec moduleRef moduleSpec specDesc specGrp specGrpRef specList tag val

textcrit: app listWit witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element pron
{
   att.global.attributes,
   att.lexicographic.attributes,
   attribute extent { data.enumerated }?,
   attribute notation { data.enumerated }?,
   macro.paraContent}
```

Example

```
<entry>
  <form>
    <orth>obverse</orth>
    <pron>'äb-`ərs</pron>,
  <pron  extent="prefix">äb-`</pron>, <pron  extent="prefix">əb-`</pron>
  </form>
  <gramGrp>
    <pos>n</pos>
  </gramGrp>
</entry>
```

<provenance> contains any descriptive or other information concerning a single identifiable episode during the history of a manuscript or manuscript part, after its creation but before its acquisition.

Module msdescription — *10. Manuscript Description*

Attributes att.datableatt.datable.w3c (*@period*, *@when*, *@notBefore*, *@notAfter*, *@from*, *@to*)
att.datable.iso (*@when-iso*, *@notBefore-iso*, *@notAfter-iso*, *@from-iso*, *@to-iso*)

Used by history

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address bibl biblStruct binaryObject cb choice cit corr date del desc distinct email emph expan foreign gap gloss graphic hi index l label lb lg list listBibl measure measureGrp mentioned milestone name note num orig p pb ptr q quote ref reg rs said sic soCalled sp stage term time title unclear

dictionaries: lang oRef oVar pRef pVar

drama: camera caption castList move sound tech view

figures: figure formula table

gaiji: g

header: biblFull

iso-fs: fLib fs fvLib

linking: ab alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material msDesc origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName listNym listOrg listPerson listPlace nameLink offset orgName persName placeName region roleName settlement state surname

nets: eTree forest forestGrp graph tree

spoken: incident kinesic pause shift u vocal writing

tagdocs: att classSpec code eg egXML elementSpec gi ident listRef macroSpec moduleRef moduleSpec schemaSpec specDesc specGrp specGrpRef specList tag val

textcrit: app listWit witDetail

textstructure: floatingText

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element provenance
{
   att.global.attributes,
   att.datable.w3c.attributes,
   att.datable.iso.attributes,
   macro.specialPara}
```

Example

```
<provenance>Listed as the property of Lawrence Sterne in 1788.</provenance>
<provenance>Sold at Sothebys in 1899.</provenance>
```

<ptr/> (pointer) defines a pointer to another location.

Module core — *3. Elements Available in All TEI Documents*

Attributes att.pointing (*@type*, *@evaluate*) att.declaring (*@decls*)

@target specifies the destination of the pointer by supplying one or more URI References

Status **Required**

Datatype 1–∞ occurrences of `data.pointer` separated by whitespace

Values One or more syntactically valid URI references, separated by whitespace. Because whitespace is used to separate URIs, no whitespace is permitted inside a single URI. If a whitespace character is required in a URI, it should be escaped with the normal mechanism, e.g. `TEI%20Consortium`.

@cRef (canonical reference) specifies the destination of the pointer by supplying a canonical reference from a scheme defined in a <refsDecl> element in the TEI header

Status **Required**

Datatype 1–∞ occurrences of `data.word` separated by whitespace

Values the result of applying the algorithm for the resolution of canonical references (described in section *16.2.5. Canonical References*) should be a valid URI reference to the intended target

Note The <refsDecl> to use may be indicated with the **decls** attribute.Currently these Guidelines only provide for a single canonical reference to be encoded on any given <ptr> element.

Used by altGrp joinGrp linkGrp listRef model.ptrLike

May contain Empty element

Declaration

```
element ptr
{
   att.global.attributes,
   att.pointing.attributes,
   att.declaring.attributes,
   (
      attribute target { list { data.pointer+ } }
    | attribute cRef { list { data.word+ } }
   ),
   empty
}
```

Example

```
<ptr target="#p143 #p144"/>
<ptr target="http://www.tei-c.org"/>
```

Note The **target** and **cRef** attributes are mutually exclusive.

<pubPlace> (publication place) contains the name of the place where a bibliographic item was published.

Module core — *3. Elements Available in All TEI Documents*

Attributes att.naming (*@key*, *@ref*, *@nymRef*)

Used by docImprint model.imprintPart model.publicationStmtPart

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address binaryObject cb choice corr date del distinct email emph expan foreign gap gloss graphic hi index lb measure measureGrp mentioned milestone name note num orig pb ptr ref reg rs sic soCalled term time title unclear

dictionaries: lang oRef oVar pRef pVar

figures: formula

gaiji: g

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident specDesc specList tag val

textcrit: app witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element pubPlace
{
   att.global.attributes,
   att.naming.attributes,
   macro.phraseSeq}
```

Example

```
<publicationStmt>
  <publisher>Oxford University Press</publisher>
  <pubPlace>Oxford</pubPlace>
  <date>1989</date>
</publicationStmt>
```

<publicationStmt> (publication statement) groups information concerning the publication or distribution of an electronic or other text.

Module header — *2. The TEI Header*

Attributes Global attributes only

Used by biblFull fileDesc

May contain

core: address date p pubPlace publisher

header: authority availability distributor idno

linking: ab

Declaration

```
element publicationStmt
{
   att.global.attributes,
   ( model.pLike+ | model.publicationStmtPart+ )
}
```

Example

```
<publicationStmt>
  <publisher>C. Muquardt </publisher>
  <pubPlace>Bruxelles & Leipzig</pubPlace>
  <date  when="1846"/>
</publicationStmt>
```

Example

```
<publicationStmt>
  <publisher>Chadwyck Healey</publisher>
  <pubPlace>Cambridge</pubPlace>
  <availability>
    <p>Available under licence only</p>
  </availability>
  <date  when="1992">1992</date>
</publicationStmt>
```

Note Although not enforced by the schemas, it is a requirement for TEI conformance that information about publication place, address, identifier, availability, and date be given in that order, following the name of the publisher, distributor, or authority concerned

---

<publisher> provides the name of the organization responsible for the publication or distribution of a bibliographic item.

Module core — *3. Elements Available in All TEI Documents*

Attributes Global attributes only

Used by docImprint model.imprintPart model.publicationStmtPart

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address binaryObject cb choice corr date del distinct email emph expan foreign gap gloss graphic hi index lb measure measureGrp mentioned milestone name note num orig pb ptr ref reg rs sic soCalled term time title unclear

dictionaries: lang oRef oVar pRef pVar

figures: formula

gaiji: g

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident specDesc specList tag val

textcrit: app witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element publisher { att.global.attributes, macro.phraseSeq }
```

Example

```
<imprint>
  <pubPlace>Oxford</pubPlace>
  <publisher>Clarendon Press</publisher>
  <date>1987</date>
</imprint>
```

Note Use the full form of the name by which a company is usually referred to, rather than any abbreviation of it which may appear on a title page

---

<purpose> characterizes a single purpose or communicative function of the text.

Module corpus — *15. Language Corpora*

Attributes In addition to global attributes

@type specifies a particular kind of purpose.

Status Optional

Datatype `data.enumerated`

Suggested values include: **persuade** didactic, advertising, propaganda, etc.

express self expression, confessional, etc.

inform convey information, educate, etc.

entertain amuse, entertain, etc.

@degree specifies the extent to which this purpose predominates.

Status Optional

Datatype `data.certainty`

Note Values should be interpreted as follows.

high this purpose is predominant

medium this purpose is intermediate

low this purpose is weak

unknown extent unknown

Used by textDesc

May contain

analysis: interp interpGrp span spanGrp

certainty: certainty respons

core: abbr address cb choice date distinct email emph expan foreign gap gloss index lb measure measureGrp mentioned milestone name note num pb ptr ref rs soCalled term time title

dictionaries: lang

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident tag val

textcrit: witDetail

transcr: addSpan am damageSpan delSpan ex fw handShift space subst

Declaration

```
element purpose
{
   att.global.attributes,
   attribute type
   {
      "persuade" | "express" | "inform" | "entertain" | xsd:Name
   }?,
   attribute degree { data.certainty }?,
   macro.phraseSeq.limited}
```

Example

```
<purpose type="persuade" degree="high"/>
<purpose type="entertain" degree="low"/>
```

Note Usually empty, unless some further clarification of the type attribute is needed, in which case it may contain running prose

---

<q> (separated from the surrounding text with quotation marks) contains material which is marked as (ostensibly) being somehow different than the surrounding text, for any one of a variety of reasons including, but not limited to: direct speech or thought, technical terms or jargon, authorial distance, quotations from elsewhere, and passages that are mentioned but not used.

Module core — *3. Elements Available in All TEI Documents*

Attributes att.ascribed (*@who*)

@type may be used to indicate whether the offset passage is spoken or thought, or to characterize it more finely.

Status Required when applicable

Datatype `data.enumerated`

Suggested values include: **spoken** representation of speech

thought representation of thought, e.g. internal monologue

written quotation from a written source

soCalled authorial distance

foreign (foreign words)

distinct (linguistically distinct)

term (technical term)

emph (rhetorically emphasized)

mentioned refering to itself, not its normal referant

Used by model.qLike

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address bibl biblStruct binaryObject cb choice cit corr date del desc distinct email emph expan foreign gap gloss graphic hi index l label lb lg list listBibl measure measureGrp mentioned milestone name note num orig p pb ptr q quote ref reg rs said sic soCalled sp stage term time title unclear

dictionaries: lang oRef oVar pRef pVar

drama: camera caption castList move sound tech view

figures: figure formula table

gaiji: g

header: biblFull

iso-fs: fLib fs fvLib

linking: ab alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material msDesc origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName listNym listOrg listPerson listPlace nameLink offset orgName persName placeName region roleName settlement state surname

nets: eTree forest forestGrp graph tree

spoken: incident kinesic pause shift u vocal writing

tagdocs: att classSpec code eg egXML elementSpec gi ident listRef macroSpec moduleRef moduleSpec schemaSpec specDesc specGrp specGrpRef specList tag val

textcrit: app listWit witDetail

textstructure: floatingText

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element q
{
   att.global.attributes,
   att.ascribed.attributes,
   attribute type
   {
      "spoken"
    | "thought"
    | "written"
    | "soCalled"
    | "foreign"
    | "distinct"
    | "term"
    | "emph"
    | "mentioned"
    | xsd:Name
   }?,
   macro.specialPara}
```

Example

```
It is spelled <q>Tübingen</q> – to enter
the letter <q>u</q> with an umlaut hold down the <q>option</q> key and press
<q>0 0 f c</q>
```

Note May be used to indicate that a passage is distinguished from the surrounding text by quotation marks for reasons concerning which no claim is made. When used in this manner, <q> may be thought of as syntactic sugar for <hi> with a value of **rend** that indicates the use of quotation marks.

<quotation> specifies editorial practice adopted with respect to quotation marks in the original.

Module header — *2. The TEI Header*

Attributes att.declarable (*@default*)

@marks (quotation marks) indicates whether or not quotation marks have been retained as content within the text.

Status Optional

Legal values are: **none** no quotation marks have been retained

some some quotation marks have been retained

all all quotation marks have been retained **[Default]**

@form specifies how quotation marks are indicated within the text.

Status Optional

Note The **form** attribute is deprecated. Although retained for compatibility, this attribute will be removed at a subsequent release.

Used by model.editorialDeclPart

May contain

core: p

linking: ab

Declaration

```
element quotation
{
   att.global.attributes,
   att.declarable.attributes,
   attribute marks { "none" | "some" | "all" }?,
   attribute form { text }?,
   model.pLike+
}
```

Example

```
<quotation  marks="none">
  <p>No quotation marks have been retained.  Instead, the
  <att>rend</att> attribute on the <gi>q</gi> element is used
     to specify what kinds of quotation mark was used, according
     to the following list:
  <list  type="gloss">
     <label>dq</label>
     <item>double quotes, open and close</item>
     <label>sq</label>
     <item>single quotes, open and close</item>
     <label>dash</label>
     <item>long dash open, no close</item>
     <label>dg</label>
     <item>double guillemets, open and close</item>
   </list>
  </p>
</quotation>
```

Example

```
<quotation  marks="all">
  <p>All quotation marks are retained in the text and
     are represented by appropriate Unicode characters.</p>
</quotation>
```

<quote> (quotation) contains a phrase or passage attributed by the narrator or author to some agency external to the text.

Module core — *3. Elements Available in All TEI Documents*

Attributes att.typed (*@type*, *@subtype*) att.msExcerpt (*@defective*)

Used by model.quoteLike

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address bibl biblStruct binaryObject cb choice cit corr date del desc distinct email emph expan foreign gap gloss graphic hi index l label lb lg list listBibl measure measureGrp mentioned milestone name note num orig p pb ptr q quote ref reg rs said sic soCalled sp stage term time title unclear

dictionaries: lang oRef oVar pRef pVar

drama: camera caption castList move sound tech view

figures: figure formula table

gaiji: g

header: biblFull

iso-fs: fLib fs fvLib

linking: ab alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material msDesc origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName listNym listOrg listPerson listPlace nameLink offset orgName persName placeName region roleName settlement state surname

nets: eTree forest forestGrp graph tree

spoken: incident kinesic pause shift u vocal writing

tagdocs: att classSpec code eg egXML elementSpec gi ident listRef macroSpec moduleRef moduleSpec schemaSpec specDesc specGrp specGrpRef specList tag val

textcrit: app listWit witDetail

textstructure: floatingText

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element quote
{
   att.global.attributes,
   att.typed.attributes,
   att.msExcerpt.attributes,
   macro.specialPara}
```

Example

```
Lexicography has shown little sign of being affected by the
work of followers of J.R. Firth, probably best summarized
in his slogan, <quote>You shall know a word by the company it
keeps</quote>
<ref>(Firth, 1957)</ref>
```

Source: [96]

Note If a bibliographic citation is supplied for the source of a quotation, the two may be grouped using the <cit> element.

<rdg> (reading) contains a single reading within a textual variation.

Module textcrit — *12. Critical Apparatus*

Attributes att.textCritical (*@wit*, *@type*, *@cause*, *@varSeq*, *@resp*, *@hand*)

Used by model.rdgLike

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address bibl biblStruct binaryObject cb choice cit corr date del desc distinct email emph expan foreign gap gloss graphic hi index label lb list listBibl measure measureGrp mentioned milestone name note num orig pb ptr q quote ref reg rs said sic soCalled stage term time title unclear

dictionaries: lang oRef oVar pRef pVar

drama: camera caption castList move sound tech view

figures: figure formula table

gaiji: g

header: biblFull

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material msDesc origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName listNym listOrg listPerson listPlace nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att classSpec code eg egXML elementSpec gi ident listRef macroSpec moduleRef moduleSpec specDesc specGrp specGrpRef specList tag val

textcrit: app lacunaEnd lacunaStart listWit wit witDetail witEnd witStart

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element rdg
{
   att.global.attributes,
   att.textCritical.attributes,
   (
      text
    | model.gLike    | model.phrase    | model.inter    | model.global    | model.rdgPart   )*
}
```

Example

```
<rdg  wit="#Ra2">Eryment</rdg>
```

<rdgGrp> (reading group) within a textual variation, groups two or more readings perceived to have a genetic relationship or other affinity.

Module textcrit — *12. Critical Apparatus*

Attributes att.textCritical (*@wit*, *@type*, *@cause*, *@varSeq*, *@resp*, *@hand*)

Used by app rdgGrp

May contain

textcrit: lem rdg rdgGrp wit

Declaration

```
element rdgGrp
{
   att.global.attributes,
   att.textCritical.attributes,
   ( ( ( rdgGrp, wit? ) | ( ( lem, wit? )?, ( model.rdgLike, wit? ) )* )+ )
}
```

Example

```
<app>
  <lem  wit="#El #Ra2">though</lem>
  <rdgGrp  type="orthographic">
    <rdg  wit="#Hg">thogh</rdg>
    <rdg  wit="#La">thouhe</rdg>
  </rdgGrp>
</app>
```

Note May contain readings and nested reading groups.Note that only one <lem> element may appear within a single apparatus entry, whether it appears outside a <rdgGrp> element or within it.

<re> (related entry) contains a dictionary entry for a lexical item related to the headword, such as a compound phrase or derived form, embedded inside a larger entry.

Module dictionaries — *9. Dictionaries*

Attributes att.lexicographic (*@expand*, *@norm*, *@split*, *@value*, *@orig*, *@location*, *@mergedIn*, *@opt*) att.typed (*@type*, *@subtype*)

Used by model.entryPart.top model.entryPart

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address binaryObject cb choice cit corr date del distinct email emph expan foreign gap gloss graphic hi index lb measure measureGrp mentioned milestone name note num orig pb ptr ref reg rs sic soCalled term time title unclear

dictionaries: def dictScrap etym form gramGrp lang oRef oVar pRef pVar re sense usg xr

figures: formula

gaiji: g

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident specDesc specList tag val

textcrit: app witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element re
{
   att.global.attributes,
   att.lexicographic.attributes,
   att.typed.attributes,
   (
      text
    | model.gLike    | sense    | model.entryPart.top    | model.phrase    | model.global   )*
}
```

Example The following example from *Webster's New Collegiate Dictionary* (Springfield, Mass.: G. & C. Merriam Company, 1975) shows a single related entry for which no definition is given, since its meaning is held to be readily derivable from the root entry:

```
<entry>
  <form>
    <orth>neu·ral</orth>
    <pron>'n(y)ůr-əl</pron>
  </form>
  <gramGrp>
    <pos>adj</pos>
  </gramGrp>
  <sense  n="1">
    <def>of, relating to, or affecting a nerve or the nervous system</def>
  </sense>
  <sense  n="2"> ... </sense>
  <re>
    <form>
      <orth>neurally</orth>
      <pron  extent="suffix">-ə-lē</pron>
    </form>
    <gramGrp>
      <pos>adv</pos>
    </gramGrp>
  </re>
</entry>
```

Example The following example from *Diccionario de la Universidad de Chicago Inglés-Español y Español-Inglés / The University of Chicago Spanish Dictionary*, Fourth Edition, compiled by Carlos Castillo and Otto F. Bond (Chicago: University of Chicago Press, 1987) shows a number of related entries embedded in the main entry. The original entry resembles the following:

> **abeja** [a·bé·xa]*f.* bee;**abejera** [a·be·xé·ra]*f.* beehive;**abejón** [a·be·xóon]*m.* drone; bumblebee;**abejorro** [a·be·xó·rro]*m.* bumble bee.

One encoding for this entry would be:

```
<entry>
  <form>
```

```
    <orth>abeja</orth>
  </form>
  <gramGrp>
    <gen>f. </gen>
  </gramGrp>
  <sense  n="1.">
    <usg  type="domain"> (ento.) </usg>
    <def> bee </def>. </sense>
  <sense  n="2.">
    <def> busy bee, hard worker </def>. </sense>
  <sense  n="3.">
    <usg  orig="A."  type="domain"> (astron.) </usg>, <def> Musca </def> — </sense>
  <re>
    <form>
      <orth  orig="a. albanila"> abeja albanila </orth>, </form>
    <sense>
      <def>mason bee</def>;</sense>
  </re>
  <re>
    <form>
      <orth  orig="a. carpintera"> abeja carpintera </orth>, </form>
    <sense>
      <def>carpenter bee </def>;</sense>
  </re>
  <re>
    <form>
      <orth  xml:id="re-o3"  orig="a. reina or maestra"> abeja reina </orth>
      <orth  mergedIn="#re-o4"> abeja maestra </orth>
    </form>
    <sense>
      <def> queen bee </def>;</sense>
  </re>
  <re>
    <form>
      <orth  xml:id="re-o4"  orig="a. neutra or obrera"> abeja neutra </orth>
      <orth  mergedIn="#re-o3"> abeja obrera </orth>
    </form>
    <sense>
      <def>worker bee</def>.</sense>
  </re>
</entry>
```

Example In the much larger Simon & Schuster Spanish-English dictionary,[1] these derived forms of *abeja* are treated as separate main entries, but there are other embedded phrases shown as<re>s in its main entry for *abeja*:

> abeja, f. 1. (ento.) bee. 2. busy bee, hard worker. 3. (astron.) A., Musca. — a. albanila, mason bee; a. carpintera, carpenter bee; a. reina or maestra, queen bee; a. neutra or obrera, worker bee.

This entry may be encoded thus:

```
<entry>
  <form>
    <orth> abeja </orth>
```

[1] Tana de Gámez, ed., *Simon and Schuster's International Dictionary* (New York: Simon and Schuster, 1973).

```
  </form>
  <gramGrp>
    <gen> f. </gen>
  </gramGrp>
  <sense  n="1.">
    <usg  type="domain"> (ento.) </usg>
    <def> bee </def>.
  </sense>
  <sense  n="2.">
    <def> busy bee, hard worker </def>. </sense>
  <sense  n="3.">
    <usg  orig="A."  type="domain"> (astron.) </usg>, <def> Musca </def> —
  </sense>
  <re>
    <form>
      <orth  orig="a. albanila"> abeja albanila </orth>,
    </form>
    <sense>
      <def> mason bee </def>; </sense>
  </re>
  <re>
    <form>
      <orth  orig="a. carpintera"> abeja carpintera </orth>,
    </form>
    <sense>
      <def> carpenter bee </def>; </sense>
  </re>
  <re>
    <form>
      <orth  xml:id="re-o1"  orig="a. reina or maestra"> abeja reina </orth>
      <orth  mergedIn="#re-o1"> abeja maestra </orth>
    </form>
    <sense>
      <def> queen bee </def>; </sense>
  </re>
  <re>
    <form>
      <orth  xml:id="re-o2"  orig="a. neutra or obrera"> abeja neutra </orth>
      <orth  mergedIn="#re-o2"> abeja obrera </orth>
    </form>
    <sense>
      <def> worker bee </def> . </sense>
  </re>
</entry>
```

Note May contain character data mixed with any other elements defined in the dictionary tag set.Identical in sub-elements to an <entry> tag, and used where a dictionary has embedded information inside one entry which could have formed a separate entry. Some authorities distinguish related entries, run-on entries, and various other types of degenerate entries; no such typology is attempted here.

<recordHist>

<recordHist> (recorded history) provides information about the source and revision status of the parent manuscript description itself.

Module msdescription — *10. Manuscript Description*

Attributes Global attributes only

Used by adminInfo

May contain

core: p

header: change

linking: ab

msdescription: source

Declaration

```
element recordHist
{
   att.global.attributes,
   ( model.pLike+ | ( source, change* ) )
}
```

Example

```
<recordHist>
  <source>
    <p>Derived from <ref  target="#IMEV">IMEV 123</ref> with additional research
        by P.M.W.Robinson</p>
  </source>
  <change  when="1999-06-23">
    <name>LDB</name> (editor)
     checked examples against DTD version 3.6
  </change>
</recordHist>
```

<recording> (recording event) details of an audio or video recording event used as the source of a spoken text, either directly or from a public broadcast.

Module spoken — *8. Transcriptions of Speech*

Attributes att.declarable (*@default*) att.durationatt.duration.w3c (*@dur*) att.duration.iso (*@dur-iso*)

@type the kind of recording.

Status Optional

Legal values are: **audio** audio recording **[Default]**

video audio and video recording

Used by broadcast recordingStmt

May contain

core: date p respStmt time

linking: ab

spoken: broadcast equipment

Declaration

```
element recording
{
   att.global.attributes,
   att.declarable.attributes,
   att.duration.w3c.attributes,
```

```
   att.duration.iso.attributes,
   attribute type { "audio" | "video" }?,
   ( model.pLike+ | model.recordingPart* )
}
```

Example

```
<recording type="audio" dur="P30M">
  <equipment>
    <p>Recorded on a Sony TR444 walkman by unknown participants; remastered
        to digital tape at <placeName>Borehamwood Studios</placeName> by
    <orgName>Transcription Services Inc</orgName>.</p>
  </equipment>
</recording>
```

Example

```
<recording type="audio" dur="P10M">
  <equipment>
    <p>Recorded from FM Radio to digital tape</p>
  </equipment>
  <broadcast>
    <bibl>
      <title>Interview on foreign policy</title>
      <author>BBC Radio 5</author>
      <respStmt>
        <resp>interviewer</resp>
        <name>Robin Day</name>
      </respStmt>
      <respStmt>
        <resp>interviewee</resp>
        <name>Margaret Thatcher</name>
      </respStmt>
      <series>
        <title>The World Tonight</title>
      </series>
      <note>First broadcast on
      <date when="1989-11-27">27 Nov 89</date>
      </note>
    </bibl>
  </broadcast>
</recording>
```

Note The **dur** attribute is used to indicate the original duration of the recording.

<recordingStmt> (recording statement) describes a set of recordings used as the basis for transcription of a spoken text.

Module spoken — *8. Transcriptions of Speech*

Attributes Global attributes only

Used by model.sourceDescPart

May contain

core: p

linking: ab

spoken: recording

Declaration

```
element recordingStmt { att.global.attributes, ( model.pLike+ | recording+ ) }
```

Example

```
<recordingStmt>
  <recording type="audio" dur="P30M">
    <respStmt>
      <resp>Location recording by</resp>
      <name>Sound Services Ltd.</name>
    </respStmt>
    <equipment>
      <p>Multiple close microphones mixed down to stereo Digital
            Audio Tape, standard play, 44.1 KHz sampling frequency</p>
    </equipment>
    <date>12 Jan 1987</date>
  </recording>
</recordingStmt>
```

Example

```
<recordingStmt>
  <p>Three
     distinct recordings made by hidden microphone in early February
     2001.</p>
</recordingStmt>
```

<ref> (reference) defines a reference to another location, possibly modified by additional text or comment.

Module core — *3. Elements Available in All TEI Documents*

Attributes att.pointing (*@type*, *@evaluate*) att.declaring (*@decls*)

@target specifies the destination of the reference by supplying one or more URI References

Status Optional

Datatype 1–∞ occurrences of `data.pointer` separated by whitespace

Values One or more syntactically valid URI references, separated by whitespace. Because whitespace is used to separate URIs, no whitespace is permitted inside a single URI. If a whitespace character is required in a URI, it should be escaped with the normal mechanism, e.g. `TEI%20Consortium`.

@cRef (canonical reference) specifies the destination of the reference by supplying a canonical reference from a scheme defined in a <refsDecl> element in the TEI header

Status Optional

Datatype 1–∞ occurrences of `data.word` separated by whitespace

Values the result of applying the algorithm for the resolution of canonical references (described in section *16.2.5. Canonical References*) should be a valid URI reference to the intended target

Note The <refsDecl> to use may be indicated with the **decls** attribute.Currently these Guidelines only provide for a single canonical reference to be encoded on any given <ref> element.

Used by model.ptrLike

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address bibl biblStruct binaryObject cb choice cit corr date del desc distinct email emph expan foreign gap gloss graphic hi index label lb list listBibl measure measureGrp mentioned milestone name note num orig pb ptr q quote ref reg rs said sic soCalled stage term time title unclear

dictionaries: lang oRef oVar pRef pVar

drama: camera caption castList move sound tech view

figures: figure formula table

gaiji: g

header: biblFull

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material msDesc origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName listNym listOrg listPerson listPlace nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att classSpec code eg egXML elementSpec gi ident listRef macroSpec moduleRef moduleSpec specDesc specGrp specGrpRef specList tag val

textcrit: app listWit witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element ref
{
   att.global.attributes,
   att.pointing.attributes,
   att.declaring.attributes,
   (
      attribute target { list { data.pointer+ } }?
    | attribute cRef { list { data.word+ } }?
   ),
   macro.paraContent}
```

Example

```
<ref
   target="http://www.natcorp.ox.ac.uk/Texts/A02.xml#s2">See especially the second sentence</ref>
See also <ref>s.v. <term>locution</term>
</ref>.>
```

Note The **target** and **cRef** attributes are mutually exclusive.

<refState/> (reference state) specifies one component of a canonical reference defined by the milestone method.

Module header — *2. The TEI Header*

Attributes In addition to global attributes

@ed (edition) indicates which edition or version the milestone applies to.

Status Optional

Datatype `data.code`

Values Any string of characters; usually a siglum conventionally used for the edition.

Note If **ed** is not specified, then any milestone tag with an appropriate **unit** attribute will be selected.

@unit indicates what kind of state is changing at this milestone.

Status **Required**

Datatype `data.enumerated`

Suggested values include: **page** page breaks in the reference edition.

column column breaks.

line line breaks.

book any units termed book, liber, etc.

poem individual poems in a collection.

canto cantos or other major sections of a poem.

stanza stanzas within a poem, book, or canto.

act acts within a play.

scene scenes within a play or act.

section sections of any kind.

absent passages not present in the reference edition.

@length specifies the fixed length of the reference component.

Status Optional

Datatype `data.count`

Values Should be a positive integer; if no value is provided, the length is unlimited and goes to the next delimiter or to the end of the value.

Note When constructing a reference, if the reference component found is of numeric type, the length is made up by inserting leading zeros; if it is not, by inserting trailing blanks. In either case, reference components are truncated if necessary at the right hand side. When seeking a reference, the length indicates the number of characters which should be compared. Values longer than this will be regarded as matching, if they start correctly.

@delim (delimiter) supplies a delimiting string following the reference component.

Status Optional

Datatype `text`

Values If a single space is used it is interpreted as whitespace.

Used by refsDecl

May contain Empty element

Declaration

```
element refState
{
   att.global.attributes,
   attribute ed { data.code }?,
   attribute unit
   {
      "page"
    | "column"
    | "line"
    | "book"
    | "poem"
    | "canto"
    | "stanza"
    | "act"
    | "scene"
    | "section"
    | "absent"
    | xsd:Name
   },
   attribute length { data.count }?,
   attribute delim { text }?,
   empty
}
```

Example

```
<refState unit="book" delim=":"/>
<refState unit="line" length="4"/>
```

<refsDecl> (references declaration) specifies how canonical references are constructed for this text.

Module header — *2. The TEI Header*

Attributes att.declarable (*@default*)

Used by model.encodingPart

May contain

core: p

header: cRefPattern refState

linking: ab

Declaration

```
element refsDecl
{
   att.global.attributes,
   att.declarable.attributes,
   ( model.pLike+ | cRefPattern+ | refState+ )
}
```

Example

```
<refsDecl>
  <cRefPattern
```

```
     matchPattern="([A-Za-z0-9]+) ([0-9]+):([0-9]+)"
     replacementPattern="#xpath(//body/div[@n='$1']/div[$2]/div3[$3])"/>
</refsDecl>
```

This example is a formal representation for the referencing scheme described informally in the following example.

Example

```
<refsDecl>
  <p>References are made up by concatenating the value for the
  <att>n</att> attribute on the highest level <gi>div</gi>
     element, followed by a space, followed by the sequential
     number of the next level <gi>div</gi> followed by a colon
     followed by the sequential number of the next (and lowest)
     level <gi>div</gi>.</p>
</refsDecl>
```

<reg> (regularization) contains a reading which has been regularized or normalized in some sense.

Module core — *3. Elements Available in All TEI Documents*

Attributes att.editLike (*@cert*, *@resp*, *@evidence*, *@source*, *@precision*) att.typed (*@type*, *@subtype*)

Used by model.pPart.transcriptional model.choicePart

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address bibl biblStruct binaryObject cb choice cit corr date del desc distinct email emph expan foreign gap gloss graphic hi index label lb list listBibl measure measureGrp mentioned milestone name note num orig pb ptr q quote ref reg rs said sic soCalled stage term time title unclear

dictionaries: lang oRef oVar pRef pVar

drama: camera caption castList move sound tech view

figures: figure formula table

gaiji: g

header: biblFull

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material msDesc origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName listNym listOrg listPerson listPlace nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att classSpec code eg egXML elementSpec gi ident listRef macroSpec moduleRef moduleSpec specDesc specGrp specGrpRef specList tag val

textcrit: app listWit witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element reg
{
   att.global.attributes,
   att.editLike.attributes,
   att.typed.attributes,
   macro.paraContent}
```

Example If all that is desired is to call attention to the fact that the copy text has been regularized, <reg> may be used alone:

```
<q>Please <reg>knock</reg> if an <reg>answer</reg>
is <reg>required</reg>
</q>
```

Example It is also possible to identify the individual responsible for the regularization, and, using the <choice> and <orig> elements, to provide both the original and regularized readings:

```
<q>Please
<choice>
   <reg  resp="#LB">knock</reg>
   <orig>cnk</orig>
  </choice> if an
<choice>
   <reg>answer</reg>
   <orig>nsr</orig>
  </choice> is
<choice>
   <reg>required</reg>
   <orig>reqd</orig>
  </choice>
</q>
```

Source: [142]

<region> contains the name of an administrative unit such as a state, province, or county, larger than a settlement, but smaller than a country.

Module namesdates — *13. Names, Dates, People, and Places*

Attributes att.naming (*@key*, *@ref*, *@nymRef*) att.typed (*@type*, *@subtype*) att.datableatt.datable.w3c (*@period*, *@when*, *@notBefore*, *@notAfter*, *@from*, *@to*) att.datable.iso (*@when-iso*, *@notBefore-iso*, *@notAfter-iso*, *@from-iso*, *@to-iso*)

Used by model.placeNamePart

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address binaryObject cb choice corr date del distinct email emph expan foreign gap gloss graphic hi index lb measure measureGrp mentioned milestone name note num orig pb ptr ref reg rs sic soCalled term time title unclear

dictionaries: lang oRef oVar pRef pVar

figures: formula

gaiji: g

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident specDesc specList tag val

textcrit: app witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element region
{
   att.global.attributes,
   att.naming.attributes,
   att.typed.attributes,
   att.datable.w3c.attributes,
   att.datable.iso.attributes,
   macro.phraseSeq}
```

Example

```
<placeName>
  <region  type="state"  n="IL">Illinois</region>
</placeName>
```

<relatedItem> contains or references some other bibliographic item which is related to the present one in some specified manner, for example as a constituent or alternative version of it.

Module core — *3. Elements Available in All TEI Documents*

Attributes att.typed (*@type*, *@subtype*)

Used by biblStruct model.biblPart

May contain

core: bibl biblStruct ptr ref

header: biblFull

msdescription: msDesc

Declaration

```
element relatedItem
{
   att.global.attributes,
   att.typed.attributes,
   ( model.biblLike | model.ptrLike )
}
```

Example

```
<biblStruct>
  <monogr>
    <author>Shirley, James</author>
    <title  type="main">The gentlemen of Venice</title>
    <imprint>
      <pubPlace>New York</pubPlace>
      <publisher>Readex Microprint</publisher>
      <date>1953</date>
    </imprint>
    <extent>1 microprint card, 23 x 15 cm.</extent>
  </monogr>
  <series>
    <title>Three centuries of drama: English, 1642–1700</title>
  </series>
  <relatedItem  type="original">
    <biblStruct>
      <monogr>
        <author>Shirley, James</author>
        <title  type="main">The gentlemen of Venice</title>
        <title  type="subordinate">a tragi-comedie presented at the private
                house in Salisbury Court by Her Majesties servants</title>
        <imprint>
          <pubPlace>London</pubPlace>
          <publisher>H. Moseley</publisher>
          <date>1655</date>
        </imprint>
        <extent>78 p.</extent>
      </monogr>
    </biblStruct>
  </relatedItem>
</biblStruct>
```

<relation> (relationship) describes any kind of relationship or linkage amongst a specified group of participants.

Module namesdates — *13. Names, Dates, People, and Places*

Attributes att.datableatt.datable.w3c (*@period*, *@when*, *@notBefore*, *@notAfter*, *@from*, *@to*) att.datable.iso (*@when-iso*, *@notBefore-iso*, *@notAfter-iso*, *@from-iso*, *@to-iso*) att.editLike (*@cert*, *@resp*, *@evidence*, *@source*, *@precision*) att.naming (*@key*, *@ref*, *@nymRef*)

@type categorizes the relationship in some respect, e.g. as social, personal or other.

Status Optional

Datatype `data.enumerated`

Suggested values include: **social** relationship concerned with social roles

personal relationship concerned with personal roles, e.g. kinship, marriage, etc. **[Default]**

other other kinds of relationship

@name supplies a name for the kind of relationship of which this is an instance.

Status **Required**

Datatype `data.enumerated`

Values an open list of application-dependent keywords

@active identifies the 'active' participants in a non-mutual relationship, or all the participants in a mutual one.

Status Optional

Datatype 1–∞ occurrences of `data.pointer` separated by whitespace

Values a list of identifier values for participant or participant groups

@mutual supplies a list of participants amongst all of whom the relationship holds equally.

Status Mandatory when applicable

Datatype 1–∞ occurrences of `data.pointer` separated by whitespace

Values a list of identifier values for participant or participant groups

@passive identifies the 'passive' participants in a non-mutual relationship.

Status Optional

Datatype 1–∞ occurrences of `data.pointer` separated by whitespace

Values a list of identifier values for participant or participant groups

Used by listNym listOrg listPerson listPlace relationGrp

May contain

core: desc

Declaration

```
element relation
{
   att.global.attributes,
   att.datable.w3c.attributes,
   att.datable.iso.attributes,
   att.editLike.attributes,
   att.naming.attributes,
   attribute type { "social" | "personal" | "other" | xsd:Name }?,
   attribute name { data.enumerated },
   (
      attribute active { list { data.pointer+ } }?
    | attribute mutual { list { data.pointer+ } }?
   ),
   attribute passive { list { data.pointer+ } }?,
   desc?
}
```

Example

```
<relation
   type="social"
   name="supervisor"
   active="#p1"
   passive="#p2 #p3 #p4"/>
```

This indicates that the person with identifier p1 is supervisor of persons p2, p3, and p4.

Example

```
<relation type="personal" name="friends" mutual="#p2 #p3 #p4"/>
```

This indicates that p2, p3, and p4 are all friends.

Note Only one of the attributes **active** and **mutual** may be supplied; the attribute **passive** may be supplied only if the attribute **active** is supplied. Not all of these constraints can be enforced in all schema languages.

<relationGrp> (relation group) provides information about relationships identified amongst people, places, and organizations, either informally as prose or as formally expressed relation links.

Module namesdates — *13. Names, Dates, People, and Places*

Attributes att.typed (*@type*, *@subtype*)

Used by listNym listOrg listPerson listPlace

May contain

core: p

linking: ab

namesdates: relation

Declaration

```
element relationGrp
{
   att.global.attributes,
   att.typed.attributes,
   ( model.pLike+ | relation+ )
}
```

Example

```
<listPerson>
  <person xml:id="p1">
<!-- data about person p1 -->
  </person>
<!-- more person elements here -->
</listPerson>
<relationGrp type="personal">
  <relation name="parent" active="#p1 #p2" passive="#p3 #p4"/>
  <relation name="spouse" mutual="#p1 #p2"/>
</relationGrp>
<relationGrp type="social">
  <relation name="employer" active="#p1" passive="#p3 #p5 #p6 #p7"/>
</relationGrp>
```

The persons with identifiers p1 and p2 are the parents of p3 and p4; they are also married to each other; p1 is the employer of p3, p5, p6, and p7.

Example

```
<relationGrp>
  <p>All speakers are members of the Ceruli family, born in Naples.</p>
</relationGrp>
```

Note May contain a prose description organized as paragraphs, or a sequence of <relation> elements.

<remarks> contains any commentary or discussion about the usage of an element, attribute, class, or entity not otherwise documented within the containing element.

Module tagdocs — *22. Documentation Elements*

Attributes att.translatable (*@version*)

Used by attDef classSpec elementSpec macroSpec moduleSpec

May contain

core: p

linking: ab

Declaration

```
element remarks
{
   att.global.attributes,
   att.translatable.attributes,
   model.pLike+
}
```

Example

```
<remarks>
  <p>This element is probably redundant.</p>
</remarks>
```

Note Contains at least one paragraph, unless it is empty.As defined in ODD, must contain paragraphs; should be special.para

---

<rendition> supplies information about the rendition or appearance of one or more elements in the source text.

Module header — *2. The TEI Header*

Attributes In addition to global attributes

@scheme identifies the language used to describe the rendition.

Status Optional

Legal values are: **css** Cascading Stylesheet Language

xslfo Extensible Stylesheet Language Formatting Objects

free Informal free text description

other A user-defined rendition description language

Used by tagsDecl

May contain

core: abbr address bibl biblStruct choice cit date desc distinct email emph expan foreign gloss label list listBibl measure measureGrp mentioned name num ptr q quote ref rs said soCalled stage term time title

dictionaries: lang

drama: camera caption castList move sound tech view

figures: figure table

header: biblFull

msdescription: catchwords depth dimensions height heraldry locus material msDesc origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName listNym listOrg listPerson listPlace nameLink offset orgName persName placeName region roleName settlement state surname

tagdocs: att classSpec code eg egXML elementSpec gi ident listRef macroSpec moduleRef moduleSpec specGrp specGrpRef tag val

textcrit: listWit

transcr: am ex handShift subst

Declaration

```
element rendition
{
   att.global.attributes,
   attribute scheme { "css" | "xslfo" | "free" | "other" }?,
   macro.limitedContent}
```

Example

```
<tagsDecl>
  <rendition  xml:id="r-center"  scheme="css">text-align: center;</rendition>
  <rendition  xml:id="r-small"  scheme="css">font-size:
     small;</rendition>
  <rendition  xml:id="r-large"  scheme="css">font-size: large;</rendition>
</tagsDecl>
```

Note The present release of these Guidelines does not specify the content of this element in any further detail. It may be used to hold a description of the default rendition to be associated with the specified element, expressed in running prose, or in some more formal language such as CSS.

<repository> contains the name of a repository within which manuscripts are stored, possibly forming part of an institution.

Module msdescription — *10. Manuscript Description*

Attributes att.naming (*@key*, *@ref*, *@nymRef*)

Used by altIdentifier msIdentifier

May contain

gaiji: g

Declaration

```
element repository
{
```

```
   att.global.attributes,
   att.naming.attributes,
   macro.xtext}
```

Example

```
<msIdentifier>
  <settlement>Oxford</settlement>
  <institution>University of Oxford</institution>
  <repository>Bodleian Library</repository>
  <idno>MS. Bodley 406</idno>
</msIdentifier>
```

<residence> (residence) describes a person's present or past places of residence.

Module namesdates — *13. Names, Dates, People, and Places*

Attributes att.datableatt.datable.w3c (*@period*, *@when*, *@notBefore*, *@notAfter*, *@from*, *@to*) att.datable.iso (*@when-iso*, *@notBefore-iso*, *@notAfter-iso*, *@from-iso*, *@to-iso*) att.editLike (*@cert*, *@resp*, *@evidence*, *@source*, *@precision*) att.naming (*@key*, *@ref*, *@nymRef*)

Used by model.persStateLike

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address binaryObject cb choice corr date del distinct email emph expan foreign gap gloss graphic hi index lb measure measureGrp mentioned milestone name note num orig pb ptr ref reg rs sic soCalled term time title unclear

dictionaries: lang oRef oVar pRef pVar

figures: formula

gaiji: g

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident specDesc specList tag val

textcrit: app witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element residence
{
   att.global.attributes,
```

```
  att.datable.w3c.attributes,
  att.datable.iso.attributes,
  att.editLike.attributes,
  att.naming.attributes,
  macro.phraseSeq}
```

Example

```
<residence>Childhood in East Africa and long term
resident of Glasgow, Scotland.</residence>
```

Example

```
<residence  notAfter="1997">Mbeni estate, Dzukumura region,
Matabele land</residence>
<residence  notBefore="1903"  notAfter="1996">
  <placeName>
    <settlement>Glasgow</settlement>
    <region>Scotland</region>
  </placeName>
</residence>
```

<resp> (responsibility) contains a phrase describing the nature of a person's intellectual responsibility.

Module core — *3. Elements Available in All TEI Documents*

Attributes Global attributes only

Used by respStmt

May contain

analysis: interp interpGrp span spanGrp

certainty: certainty respons

core: abbr address cb choice date distinct email emph expan foreign gap gloss index lb measure measureGrp mentioned milestone name note num pb ptr ref rs soCalled term time title

dictionaries: lang

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident tag val

textcrit: witDetail

transcr: addSpan am damageSpan delSpan ex fw handShift space subst

Declaration

```
element resp { att.global.attributes, macro.phraseSeq.limited }
```

Example

```
<respStmt>
  <resp>compiler</resp>
  <name>Edward Child</name>
</respStmt>
```

<respStmt> (statement of responsibility) supplies a statement of responsibility for the intellectual content of a text, edition, recording, or series, where the specialized elements for authors, editors, etc. do not suffice or do not apply.

Module core — *3. Elements Available in All TEI Documents*

Attributes Global attributes only

Used by analytic editionStmt monogr msItemStruct series seriesStmt titleStmt model.respLike model.recordingPart

May contain

core: name resp

namesdates: orgName persName

Declaration

```
element respStmt
{
   att.global.attributes,
   ( ( resp+, model.nameLike.agent+ ) | ( model.nameLike.agent+, resp+ ) )
}
```

Example

```
<respStmt>
  <resp>transcribed from original ms</resp>
  <persName>Claus Huitfeldt</persName>
</respStmt>
```

Example

```
<respStmt>
  <resp>converted to SGML encoding</resp>
  <name>Alan Morrison</name>
</respStmt>
```

<respons> (responsibility) identifies the individual(s) responsible for some aspect of the markup of particular element(s).

Module certainty — *21. Certainty and Responsibility*

Attributes In addition to global attributes

@target gives the identifier(s) of the element(s) for which some aspect of the responsibility is being assigned.

Status **Required**

Datatype 1–∞ occurrences of `data.pointer` separated by whitespace

Values one or more valid identifiers, separated by whitespace.

@locus indicates the specific aspect of the markup for which responsibility is being assigned.

Status **Required**

Datatype 1–∞ occurrences of `data.enumerated` separated by whitespace

Suggested values include: **gi** (element name) responsibility for the claim that the element is of the type indicated by the markup

location responsibility for the claim that the element begins and ends where indicated

startLoc (start location) responsibility for the claim that the element begins where indicated

endLoc (end location) responsibility for the claim that the element ends where indicated

attrName (attribute name) responsibility for the claim that the name attribute has the value given in the markup

transcribedContent responsibility for the transcription of the element content

suppliedContent responsibility for the contents supplied by the encoder (corrections, expansions of abbreviations, etc.)

@resp (responsible party) identifies the individual or agency responsible for the indicated aspect of the electronic text.

Status **Required**

Datatype `data.pointer`

Values a pointer to one of the identifiers declared in the document header, associated with a person asserted as responsible for some aspect of the text's creation, transcription, editing, or encoding

Used by model.global.meta

May contain

core: desc gloss

tagdocs: altIdent equiv

Declaration

```
element respons
{
   att.global.attributes,
   attribute target { list { data.pointer+ } },
   attribute locus
   {
      list
      {
         (
            "gi"
          | "location"
          | "startLoc"
          | "endLoc"
          | "attrName"
          | "transcribedContent"
          | "suppliedContent"
          | xsd:Name
         )+
      }
```

```
   },
   attribute resp { data.pointer },
   model.glossLike*
}
```

Example

```
<respons target="#p1" locus="gi location" resp="#encoder1"/>
<respons target="#p2" locus="rend" resp="#encoder2"/>
<list type="encoders">
  <item xml:id="encoder1"/>
  <item xml:id="encoder2"/>
</list>
```

Note The <respons> element is designed for cases in which fine-grained information about specific aspects of the markup of a text is desirable for whatever reason. Global responsibility for certain aspects of markup is usually more simply indicated in the TEI header, using the <respStmt> element within the title statement, edition statement, or change log.

<restore> indicates restoration of text to an earlier state by cancellation of an editorial or authorial marking or instruction.

Module transcr — *11. Representation of Primary Sources*

Attributes att.transcriptional (*@hand*, *@status*, *@seq*) att.editLike (*@cert*, *@resp*, *@evidence*, *@source*, *@precision*) att.typed (*@type*, *@subtype*)

Used by model.pPart.transcriptional

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address bibl biblStruct binaryObject cb choice cit corr date del desc distinct email emph expan foreign gap gloss graphic hi index label lb list listBibl measure measureGrp mentioned milestone name note num orig pb ptr q quote ref reg rs said sic soCalled stage term time title unclear

dictionaries: lang oRef oVar pRef pVar

drama: camera caption castList move sound tech view

figures: figure formula table

gaiji: g

header: biblFull

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material msDesc origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName listNym listOrg listPerson listPlace nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att classSpec code eg egXML elementSpec gi ident listRef macroSpec moduleRef moduleSpec specDesc specGrp specGrpRef specList tag val

textcrit: app listWit witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element restore
{
   att.global.attributes,
   att.transcriptional.attributes,
   att.editLike.attributes,
   att.typed.attributes,
   macro.paraContent}
```

Example

```
For I hate this
<restore  hand="#dhl"  type="marginalStetNote">
  <del>my</del>
</restore> body
```

Note On this element, the **type** attribute indicates the action cancelled by the restoration. Its value should be the name of the tag contained within the <restore> element which is cancelled by the restoration. Most often, this will be <del>, but might also be <hi>, etc. In cases of simple nesting of a single cancelled action within the <restore> element this attribute will not be necessary.

<revisionDesc> (revision description) summarizes the revision history for a file.

Module header — *2. The TEI Header*

Attributes Global attributes only

Used by teiHeader

May contain

core: list

header: change

Declaration

```
element revisionDesc { att.global.attributes, ( list | change+ ) }
```

Example

```
<revisionDesc>
  <change  when="1991-11-11"> EMB deleted chapter 10
  </change>
</revisionDesc>
```

Note Record changes with most recent changes at the top of the list.

<rhyme> marks the rhyming part of a metrical line.

Module verse — *6. Verse*

Attributes att.typed (*@type*, *@subtype*)

@label provides a label to identify which part of a rhyme scheme this rhyming string instantiates.

Status Recommended

Datatype `data.word`

Values usually contains a single letter.

Note Within a particular scope, all <rhyme> elements with the same value for their **label** attribute are assumed to rhyme with each other. The scope is defined by the nearest ancestor element for which the **rhyme** attribute has been supplied.

Used by model.lPart

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address bibl biblStruct binaryObject cb choice cit corr date del desc distinct email emph expan foreign gap gloss graphic hi index label lb list listBibl measure measureGrp mentioned milestone name note num orig pb ptr q quote ref reg rs said sic soCalled stage term time title unclear

dictionaries: lang oRef oVar pRef pVar

drama: camera caption castList move sound tech view

figures: figure formula table

gaiji: g

header: biblFull

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material msDesc origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName listNym listOrg listPerson listPlace nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att classSpec code eg egXML elementSpec gi ident listRef macroSpec moduleRef moduleSpec specDesc specGrp specGrpRef specList tag val

textcrit: app listWit witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element rhyme
{
   att.global.attributes,
   att.typed.attributes,
```

```
attribute label { data.word }?,
macro.paraContent}
```

Example

```
<lg  rhyme="abababcc">
  <l>'Tis pity learned virgins ever <rhyme  label="a">wed</rhyme>
  </l>
  <l>With persons of no sort of edu<rhyme  label="b">cation</rhyme>,</l>
  <l>Or gentlemen, who, though well born and <rhyme  label="a">bred</rhyme>,</l>
  <l>Grow tired of scientific conver<rhyme  label="b">sation</rhyme>:</l>
  <l>I don't choose to say much on this <rhyme  label="a">head</rhyme>,</l>
  <l>I'm a plain man, and in a single <rhyme  label="b">station</rhyme>,</l>
  <l>But — Oh! ye lords of ladies inte<rhyme  label="c">llectual</rhyme>,</l>
  <l>Inform us truly, have they not hen-<rhyme  label="a">peck'd you all</rhyme>?</l>
</lg>
```

Source: [26]

<role> the name of a dramatic role, as given in a cast list.

Module drama — *7. Performance Texts*

Attributes Global attributes only

Used by model.castItemPart

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address binaryObject cb choice corr date del distinct email emph expan foreign gap gloss graphic hi index lb measure measureGrp mentioned milestone name note num orig pb ptr ref reg rs sic soCalled term time title unclear

dictionaries: lang oRef oVar pRef pVar

figures: formula

gaiji: g

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident specDesc specList tag val

textcrit: app witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element role { att.global.attributes, macro.phraseSeq }
```

Example

```
<role  xml:id="jt">Joan Trash</role>
<roleDesc>A Ginger-bread-woman</roleDesc>
```

Note It is important to assign a meaningful ID attribute to the <role> element, since this ID is referred to by **who** attributes on many other elements.

<roleDesc> (role description) describes a character's role in a drama.

Module drama — *7. Performance Texts*

Attributes Global attributes only

Used by castGroup model.castItemPart

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address binaryObject cb choice corr date del distinct email emph expan foreign gap gloss graphic hi index lb measure measureGrp mentioned milestone name note num orig pb ptr ref reg rs sic soCalled term time title unclear

dictionaries: lang oRef oVar pRef pVar

figures: formula

gaiji: g

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident specDesc specList tag val

textcrit: app witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element roleDesc { att.global.attributes, macro.phraseSeq }
```

Example

```
<roleDesc>gentlemen of leisure</roleDesc>
```

<roleName> contains a name component which indicates that the referent has a particular role or position in society, such as an official title or rank.

Module namesdates — *13. Names, Dates, People, and Places*

Attributes att.personal (*@full*, *@sort*) att.naming (*@key*, *@ref*, *@nymRef*) att.typed (*@type*, *@subtype*)

Used by model.persNamePart

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address binaryObject cb choice corr date del distinct email emph expan foreign gap gloss graphic hi index lb measure measureGrp mentioned milestone name note num orig pb ptr ref reg rs sic soCalled term time title unclear

dictionaries: lang oRef oVar pRef pVar

figures: formula

gaiji: g

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident specDesc specList tag val

textcrit: app witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element roleName
{
   att.global.attributes,
   att.personal.attributes,
   att.naming.attributes,
   att.typed.attributes,
   macro.phraseSeq}
```

Example

```
<persName>
  <forename>William</forename>
  <surname>Poulteny</surname>
  <roleName>Earl of Bath</roleName>
</persName>
```

Note A <roleName> may be distinguished from an <addName> by virtue of the fact that, like a title, it typically exists independently of its holder.

<root> (root node) represents the root node of a tree.

Module nets — *19. Graphs, Networks, and Trees*

Attributes In addition to global attributes

@value provides the value of the root, which is a feature structure or other analytic element.

Status Required when applicable

Datatype `data.pointer`

Values A valid identifier of a feature structure or other analytic element.

@children provides a list of identifiers of the elements which are the children of the root node.

Status **Required**

Datatype 1–∞ occurrences of `data.pointer` separated by whitespace

Values A list of valid identifiers.

Note If the root has no children (i.e., the tree is 'trivial'), then the **children** attribute must be omitted. For technical reasons, it cannot be specified as **<root children="">**.

@ord (ordered) indicates whether or not the root is ordered.

Status Required when applicable

Datatype `data.xTruthValue`

Note The value **true** indicates that the children of the root are ordered, whereas **false** indicates the are unordered.Use if and only if **ord** is specified as **partial** on the <tree> element and the root has more than one child.

@outDegree gives the out degree of the root, the number of its children.

Status Optional

Datatype `data.count`

Values A nonnegative integer.

Note The in degree of the root is always 0.

Used by tree

May contain

core: label

Declaration

```
element root
{
   att.global.attributes,
   attribute value { data.pointer }?,
   attribute children { list { data.pointer+ } },
   attribute ord { data.xTruthValue }?,
   attribute outDegree { data.count }?,
   label?
}
```

Example

```
<root xml:id="vp1" children="#vb1 #pn1" outDegree="2">
  <label>VP</label>
</root>
<leaf xml:id="vb1"/>
<leaf xml:id="pn1"/>
```

<row> contains one row of a table.

Module figures — *14. Tables, Formulæ, and Graphics*

Attributes att.tableDecoration (*@role*, *@rows*, *@cols*)

Used by table

May contain

figures: cell

Declaration

```
element row { att.global.attributes, att.tableDecoration.attributes, cell+ }
```

Example

```
<row role="data">
  <cell role="label">Classics</cell>
  <cell>Idle listless and unimproving</cell>
</row>
```

<rs> (referencing string) contains a general purpose name or referring string.

Module core — *3. Elements Available in All TEI Documents*

Attributes att.naming (*@key*, *@ref*, *@nymRef*)

@type indicates more specifically the object referred to by the referencing string. Values might include **person**, **place**, **ship**, **element** etc.

Status Mandatory when applicable

Datatype `data.enumerated`

Values Any string of characters.

Used by model.nameLike

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address binaryObject cb choice corr date del distinct email emph expan foreign gap gloss graphic hi index lb measure measureGrp mentioned milestone name note num orig pb ptr ref reg rs sic soCalled term time title unclear

dictionaries: lang oRef oVar pRef pVar

figures: formula

gaiji: g

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident specDesc specList tag val

textcrit: app witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element rs
{
   att.global.attributes,
   att.naming.attributes,
   attribute type { data.enumerated }?,
   macro.phraseSeq}
```

Example

```
<q>My dear <rs  type="person">Mr. Bennet</rs>, </q>
said <rs  type="person">his lady</rs> to him one day,

<q>have you heard that <rs  type="place">Netherfield
    Park</rs> is let at last?</q>
```

<rubric> contains the text of any rubric or heading attached to a particular manuscript item, that is, a string of words through which a manuscript signals the beginning of a text division, often with an assertion as to its author and title, which is in some way set off from the text itself, usually in red ink, or by use of different size or type of script, or some other such visual device.

Module msdescription — *10. Manuscript Description*

Attributes att.typed (*@type*, *@subtype*)

Used by msItemStruct model.msItemPart

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address binaryObject cb choice corr date del distinct email emph expan foreign gap gloss graphic hi index lb measure measureGrp mentioned milestone name note num orig pb ptr ref reg rs sic soCalled term time title unclear

dictionaries: lang oRef oVar pRef pVar

figures: formula

gaiji: g

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident specDesc specList tag val

textcrit: app witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element rubric { att.global.attributes, att.typed.attributes, macro.phraseSeq }
```

Example

```
<rubric>Nu koma Skyckiu Rym<expan>ur</expan>.</rubric>
<rubric>Incipit liber de consciencia humana a beatissimo Bernardo editus.</rubric>
<rubric>
  <locus>16. f. 28v in margin: </locus>Dicta Cassiodori
</rubric>
```

<s> (s-unit) contains a sentence-like division of a text.

Module analysis — *17. Simple Analytic Mechanisms*

Attributes att.segLike (*@type*, *@function*, *@part*) att.metrical (*@met*, *@real*, *@rhyme*)

Used by model.segLike

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address binaryObject cb choice corr date del distinct email emph expan foreign gap gloss graphic hi index lb measure measureGrp mentioned milestone name note num orig pb ptr ref reg rs sic soCalled term time title unclear

dictionaries: lang oRef oVar pRef pVar

figures: formula

gaiji: g

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident specDesc specList tag val

textcrit: app witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element s
{
   att.global.attributes,
   att.segLike.attributes,
   att.metrical.attributes,
   macro.phraseSeq}
```

Example

```
<s>When are you leaving?</s>
<s>Tomorrow.</s>
```

Note May contain any combination of text and phrase-level elements; may not contain itself.The <s> element may be used to mark orthographic sentences, or any other segmentation of a text, provided that the segmentation is end-to-end, complete, and non-nesting.

<said> (speech or thought) indicates passages thought or spoken aloud, whether explicitly indicated in the source or not, whether directly or indirectly reported, whether by real people or fictional characters.

Module core — *3. Elements Available in All TEI Documents*

Attributes att.ascribed (*@who*)

@aloud may be used to indicate whether the quoted matter is regarded as having been vocalized or signed.

Status Required when applicable

Datatype `data.xTruthValue`

Note The value **true** indicates the encoded passage was expressed outwardly (whether spoken, signed, sung, screamed, chanted, etc.); the value **false** indicates that the encoded passage was thought, but not outwardly expressed.

@direct may be used to indicate whether the quoted matter is regarded as direct or indirect speech.

Status Required when applicable

Datatype `data.xTruthValue`

Note The value **true** indicates the speech or thought is represented directly; the value **false** that speech or thought is represented indirectly, e.g. by use of a marked verbal aspect.

Used by model.qLike

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address bibl biblStruct binaryObject cb choice cit corr date del desc distinct email emph expan foreign gap gloss graphic hi index l label lb lg list listBibl measure

measureGrp mentioned milestone name note num orig p pb ptr q quote ref reg rs said sic soCalled sp stage term time title unclear

dictionaries: lang oRef oVar pRef pVar

drama: camera caption castList move sound tech view

figures: figure formula table

gaiji: g

header: biblFull

iso-fs: fLib fs fvLib

linking: ab alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material msDesc origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName listNym listOrg listPerson listPlace nameLink offset orgName persName placeName region roleName settlement state surname

nets: eTree forest forestGrp graph tree

spoken: incident kinesic pause shift u vocal writing

tagdocs: att classSpec code eg egXML elementSpec gi ident listRef macroSpec moduleRef moduleSpec schemaSpec specDesc specGrp specGrpRef specList tag val

textcrit: app listWit witDetail

textstructure: floatingText

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element said
{
   att.global.attributes,
   att.ascribed.attributes,
   attribute aloud { data.xTruthValue }?,
   attribute direct { data.xTruthValue }?,
   macro.specialPara}
```

Example

```
<p>
  <said>Our minstrel here will warm the old man's heart
     with song, dazzle him with jewels and gold</said>, a
troublemaker simpered. <said>He'll trample on the Duke's
     camellias, spill his wine, and blunt his sword, and say
     his name begins with X, and in the end the Duke will
     say, <said>Take Saralinda, with my blessing, O lordly
        Prince of Rags and Tags, O rider of the
        sun!</said>
  </said>
</p>
```

Source: [195]

Example

```
<p>
  <said aloud="true" rend="pre(“) post(”)">Hmmm</said>,
said a small voice in his ear. <said aloud="true" rend="pre(“) post(”)">Difficult. Very
difficult.
    Plenty of courage, I see. Not a bad mind either. there's talent, oh
    my goodness, yes — and a nice thirst to prove yourself, now that's
    interesting. … So where shall I put you?</said>
</p>
<p>Harry gripped the edges of the stool and thought,
<said aloud="false" rend="italic">Not Slytherin, not
    Slytherin</said>.</p>
```

Source: [166]

<salute> (salutation) contains a salutation or greeting prefixed to a foreword, dedicatory epistle, or other division of a text, or the salutation in the closing of a letter, preface, etc.

Module textstructure — *4. Default Text Structure*

Attributes Global attributes only

Used by closer opener model.divTopPart

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address binaryObject cb choice corr date del distinct email emph expan foreign gap gloss graphic hi index lb measure measureGrp mentioned milestone name note num orig pb ptr ref reg rs sic soCalled term time title unclear

dictionaries: lang oRef oVar pRef pVar

figures: formula

gaiji: g

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident specDesc specList tag val

textcrit: app witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element salute { att.global.attributes, macro.phraseSeq }
```

Example

```
<salute>To all courteous mindes, that will voutchsafe the
readinge.</salute>
```

<samplingDecl> (sampling declaration) contains a prose description of the rationale and methods used in sampling texts in the creation of a corpus or collection.

Module header — *2. The TEI Header*

Attributes att.declarable (*@default*)

Used by model.encodingPart

May contain

core: p

linking: ab

Declaration

```
element samplingDecl
{
   att.global.attributes,
   att.declarable.attributes,
   model.pLike+
}
```

Example

```
<samplingDecl>
  <p>Samples of up to 2000 words taken at random from the beginning,
     middle, or end of each text
     identified as relevant by respondents.</p>
</samplingDecl>
```

Note This element records all information about systematic inclusion or omission of portions of the text, whether a reflection of sampling procedures in the pure sense or of systematic omission of material deemed either too difficult to transcribe or not of sufficient interest.

<schemaSpec> (schema specification) generates a TEI-conformant schema and documentation for it.

Module tagdocs — *22. Documentation Elements*

Attributes att.identified (*@ident*, *@predeclare*, *@module*, *@mode*)

@start specifies entry points to the schema, i.e. which elements are allowed to be used as the root of documents conforming to it.

Status Optional

Datatype 1–∞ occurrences of `data.name` separated by whitespace

@ns (namespace) specifies the default namespace (if any) applicable to components of the schema.

Status Optional

Datatype `data.namespace`

@prefix specifies a prefix which will be appended to all patterns relating to TEI elements. This allows for external schemas to be mixed in which have elements of the same names as the TEI.

Status Optional

Datatype `data.name`

Note Colons, although permitted inside the value, will cause an invalid schema to be generated.

@targetLang (target language) specifies which language to use when creating the objects in a schema if names for elements or attributes are available in more than one language, .

Status Optional

Datatype `data.language`

@docLang (documentation language) specifies which languages to use when creating documentation if the description for an element, attribute, class or macro is available in more than one language, .

Status Optional

Datatype 1–∞ occurrences of `data.language` separated by whitespace

Used by model.divPart

May contain

core: desc gloss

tagdocs: altIdent classSpec elementSpec equiv listRef macroSpec moduleRef moduleSpec specGrp specGrpRef

Declaration

```
element schemaSpec
{
   att.global.attributes,
   att.identified.attributes,
   attribute start { list { data.name+ } }?,
   attribute ns { data.namespace }?,
   attribute prefix { data.name }?,
   attribute targetLang { data.language }?,
   attribute docLang { list { data.language+ } }?,
   ( model.glossLike*, ( moduleRef | specGrpRef | model.oddDecl )* )
}
```

Example

```
<schemaSpec prefix="TEI_" ident="testsvg" start="TEI svg">
  <moduleRef key="header"/>
  <moduleRef key="core"/>
  <moduleRef key="tei"/>
  <moduleRef key="textstructure"/>
  <moduleRef url="svg11.rng"/>
</schemaSpec>
```

Note A schema combines references to modules or specification groups with other atomic declarations. The processing of a schema element must resolve any conflicts amongst the declarations it contains or references. Different ODD processors may generate schemas and documentation using different concrete syntaxes.

<scriptStmt> (script statement) contains a citation giving details of the script used for a spoken text.

Module spoken — *8. Transcriptions of Speech*

Attributes att.declarable (*@default*)

Used by model.sourceDescPart

May contain

core: bibl biblStruct p

header: biblFull

linking: ab

msdescription: msDesc

Declaration

```
element scriptStmt
{
   att.global.attributes,
   att.declarable.attributes,
   ( model.pLike+ | model.biblLike )
}
```

Example

```
<scriptStmt>
  <bibl>
    <author>Craig Warner</author>
    <title>Strangers on a Train</title>
    <title  type="sub">Based on the novel by Patricia Highsmith</title>
    <edition>French's acting edition</edition>
    <idno  type="isbn">978 0 573 01972 2</idno>
    <publisher>Samuel French Ltd</publisher>
  </bibl>
</scriptStmt>
```

<seal> contains a description of one seal or similar attachment applied to a manuscript.

Module msdescription — *10. Manuscript Description*

Attributes att.typed (*@type*, *@subtype*) att.datableatt.datable.w3c (*@period*, *@when*, *@notBefore*, *@notAfter*, *@from*, *@to*) att.datable.iso (*@when-iso*, *@notBefore-iso*, *@notAfter-iso*, *@from-iso*, *@to-iso*)

@contemporary specifies whether or not the seal is contemporary with the item to which it is affixed

Status Optional

Datatype `data.xTruthValue`

Used by sealDesc

May contain

core: p

linking: ab

msdescription: decoNote

Declaration

```
element seal
{
   att.global.attributes,
   att.typed.attributes,
   att.datable.w3c.attributes,
   att.datable.iso.attributes,
   attribute contemporary { data.xTruthValue }?,
   ( model.pLike | decoNote )+
}
```

Example

```
<seal  n="2"  type="pendant"  subtype="cauda_duplex">
  <p>The seal of <name>Jens Olufsen</name> in black wax.
     (<ref>DAS 1061</ref>). Legend: <q>S IOHANNES OLAVI</q>.
     Parchment tag on which is written: <q>Woldorp Iohanne G</q>.</p>
</seal>
```

<sealDesc> (seal description) describes the seals or other external items attached to a manuscript, either as a series of paragraphs or as a series of distinct <seal> elements, possibly with additional <decoNote>s.

Module msdescription — *10. Manuscript Description*

Attributes Global attributes only

Used by model.physDescPart

May contain

core: p

linking: ab

msdescription: decoNote seal

Declaration

```
element sealDesc
{
   att.global.attributes,
   ( model.pLike+ | ( decoNote | seal )+ )
}
```

<secFol> (second folio) The word or words taken from a fixed point in a codex (typically the beginning of the second leaf) in order to provide a unique identifier for it.

Module msdescription — *10. Manuscript Description*

Attributes Global attributes only

Used by model.pPart.msdesc

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address binaryObject cb choice corr date del distinct email emph expan foreign gap gloss graphic hi index lb measure measureGrp mentioned milestone name note num orig pb ptr ref reg rs sic soCalled term time title unclear

dictionaries: lang oRef oVar pRef pVar

figures: formula

gaiji: g

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident specDesc specList tag val

textcrit: app witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element secFol { att.global.attributes, macro.phraseSeq }
```

Example

```
<secFol>(con-)versio morum</secFol>
```

<seg> (arbitrary segment) represents any segmentation of text below the 'chunk' level.

Module linking — *16. Linking, Segmentation, and Alignment*

Attributes att.segLike (*@type*, *@function*, *@part*) att.metrical (*@met*, *@real*, *@rhyme*)

@subtype provides a sub-categorization of the segment marked.

Status Optional

Datatype `data.word`

Values any string of characters.

Note The **subtype** attribute may be used to provide any classification for the <seg> elements tagged in a text suitable for the **type** given.

Used by model.segLike model.choicePart

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address bibl biblStruct binaryObject cb choice cit corr date del desc distinct email emph expan foreign gap gloss graphic hi index label lb list listBibl measure measureGrp mentioned milestone name note num orig pb ptr q quote ref reg rs said sic soCalled stage term time title unclear

dictionaries: lang oRef oVar pRef pVar

drama: camera caption castList move sound tech view

figures: figure formula table

gaiji: g

header: biblFull

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material msDesc origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName listNym listOrg listPerson listPlace nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att classSpec code eg egXML elementSpec gi ident listRef macroSpec moduleRef moduleSpec specDesc specGrp specGrpRef specList tag val

textcrit: app listWit witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element seg
{
   att.global.attributes,
   att.segLike.attributes,
   att.metrical.attributes,
   attribute subtype { data.word }?,
   macro.paraContent}
```

Example

```
<seg>When are you leaving?</seg>
<seg>Tomorrow.</seg>
```

Example

```
<s>
  <seg  rend="caps"  type="initial-cap">So father's only</seg>
glory was the ballfield.

</s>
```

Example

```
<seg  type="preamble">
  <seg>Sigmund,
  <seg  type="patronym">the son of Volsung</seg>,
     was a king in Frankish country.</seg>
  <seg>Sinfiotli was the eldest of his sons ...</seg>
  <seg>Borghild, Sigmund's wife, had a brother ... </seg>
</seg>
```

Note The <seg> element may be used at the encoder's discretion to mark any segments of the text of interest for processing. One use of the element is to mark text features for which no appropriate markup is otherwise defined. Another use is to provide an identifier for some segment which is to be pointed at by some other element — i.e. to provide a target, or a part of a target, for a <ptr> or other similar element.

<segmentation> describes the principles according to which the text has been segmented, for example into sentences, tone-units, graphemic strata, etc.

Module header — *2. The TEI Header*

Attributes att.declarable (*@default*)

Used by model.editorialDeclPart

May contain

core: p

linking: ab

Declaration

```
element segmentation
{
   att.global.attributes,
   att.declarable.attributes,
   model.pLike+
}
```

Example

```
<segmentation>
  <p>
    <gi>s</gi> elements mark orthographic sentences and
     are numbered sequentially
     within their parent <gi>div</gi> element
  </p>
</segmentation>
```

Example

```
<p>
  <gi>seg</gi> elements are used to mark functional constituents
of various types within each <gi>s</gi>; the typology used is defined
by a <gi>taxonomy</gi> element in the corpus header <gi>classDecl</gi>
</p>
```

<sense> groups together all information relating to one word sense in a dictionary entry, for example definitions, examples, and translation equivalents.

Module dictionaries — *9. Dictionaries*

Attributes att.lexicographic (*@expand*, *@norm*, *@split*, *@value*, *@orig*, *@location*, *@mergedIn*, *@opt*)

@level gives the nesting depth of this sense.

Status Optional

Datatype `data.numeric`

Values a positive integer

Used by entry hom re sense model.entryPart

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address binaryObject cb choice cit corr date del distinct email emph expan foreign gap gloss graphic hi index lb measure measureGrp mentioned milestone name note num orig pb ptr ref reg rs sic soCalled term time title unclear

dictionaries: def dictScrap etym form gramGrp lang oRef oVar pRef pVar re sense usg xr

figures: formula

gaiji: g

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident specDesc specList tag val

textcrit: app witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element sense
{
   att.global.attributes,
   att.lexicographic.attributes,
   attribute level { data.numeric }?,
   (
      text
    | model.gLike    | sense    | model.entryPart.top    | model.phrase    | model.global   )*
}
```

Example

```
<sense n="2">
  <usg type="time">Vx.</usg>
  <def>Vaillance, bravoure (spécial., au combat)</def>
  <cit type="example">
    <quote>La valeur n'attend pas le nombre des années</quote>
    <bibl>
      <author>Corneille</author>
    </bibl>
  </cit>
</sense>
```

Note May contain character data mixed with any other elements defined in the dictionary tag set.

<series> (series information) contains information about the series in which a book or other bibliographic item has appeared.

Module core — *3. Elements Available in All TEI Documents*

Attributes Global attributes only

Used by biblStruct model.biblPart

May contain

analysis: interp interpGrp span spanGrp

certainty: certainty respons

core: biblScope cb editor gap index lb milestone note pb respStmt title

gaiji: g

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp timeline

spoken: incident kinesic pause shift vocal writing

textcrit: witDetail

transcr: addSpan damageSpan delSpan fw space

Declaration

```
element series
{
   att.global.attributes,
   (
      text
    | model.gLike    | title    | editor    | respStmt    | biblScope    | model.global   )*
}
```

Example

```
<series xml:lang="de">
  <title level="s">Halbgraue Reihe zur Historischen Fachinformatik</title>
  <respStmt>
    <resp>Herausgegeben von</resp>
    <name type="person">Manfred Thaller</name>
    <name type="org">Max-Planck-Institut für Geschichte</name>
  </respStmt>
  <title level="s">Serie A: Historische Quellenkunden</title>
  <biblScope>Band 11</biblScope>
</series>
```

<seriesStmt> (series statement) groups information about the series, if any, to which a publication belongs.

Module header — *2. The TEI Header*

Attributes Global attributes only

Used by biblFull fileDesc

May contain

core: p respStmt title

header: idno

linking: ab

Declaration

```
element seriesStmt
{
   att.global.attributes,
   ( model.pLike+ | ( title+, ( idno | respStmt )* ) )
}
```

Example

```
<seriesStmt>
  <title>Machine-Readable Texts for the Study of Indian
     Literature</title>
  <respStmt>
    <resp>ed. by</resp>
    <name>Jan Gonda</name>
  </respStmt>
  <idno type="vol">1.2</idno>
  <idno type="ISSN">0 345 6789</idno>
</seriesStmt>
```

<set> (setting) contains a description of the setting, time, locale, appearance, etc., of the action of a play, typically found in the front matter of a printed performance text (not a stage direction).

Module drama — *7. Performance Texts*

Attributes Global attributes only

Used by model.frontPart.drama

May contain

analysis: interp interpGrp span spanGrp

certainty: certainty respons

core: bibl biblStruct cb cit desc gap head index l label lb lg list listBibl milestone note p pb q quote said sp stage

dictionaries: entry entryFree superEntry

drama: camera caption castList move sound tech view

figures: figure table

header: biblFull

iso-fs: fLib fs fvLib

linking: ab alt altGrp anchor join joinGrp link linkGrp timeline

msdescription: msDesc

namesdates: listNym listOrg listPerson listPlace

nets: eTree forest forestGrp graph tree

spoken: incident kinesic pause shift u vocal writing

tagdocs: classSpec eg egXML elementSpec listRef macroSpec moduleRef moduleSpec schemaSpec specGrp specGrpRef

textcrit: listWit witDetail

textstructure: floatingText

transcr: addSpan damageSpan delSpan fw space

Declaration

```
element set
{
   att.global.attributes,
   ( ( model.headLike | model.global )*, ( ( model.common ), model.global* )* )
}
```

Example

```
<set>
  <p>The action takes place on February 7th between the hours of noon
     and six in the afternoon, close to the Trenartha Tin Plate Works,
     on the borders of England and Wales, where a strike has been in
     progress throughout the winter.</p>
</set>
```

Example

```
<set>
  <head>SCENE</head>
  <p>A Sub-Post Office on a late autumn evening</p>
</set>
```

Example

```
<front>
<!-- <titlePage>, <div type="Dedication">, etc. -->
  <set>
    <list  type="gloss">
      <label>TIME</label>
      <item>1907</item>
      <label>PLACE</label>
      <item>East Coast village in England</item>
    </list>
  </set>
</front>
```

Note Contains paragraphs or phrase level tags.This element should not be used outside the front matter; for similar contextual descriptions within the body of the text, use the <stage> element.

<setting> describes one particular setting in which a language interaction takes place.

Module corpus — *15. Language Corpora*

Attributes att.ascribed (*@who*)

Used by settingDesc

May contain

core: date name p time

corpus: activity locale

linking: ab

namesdates: orgName persName

Declaration

```
element setting
{
   att.global.attributes,
   att.ascribed.attributes,
   (
      model.pLike+
    | ( model.nameLike.agent | model.dateLike | model.settingPart )*
   )
}
```

Example

```
<setting>
  <name>New York City, US</name>
  <date>1989</date>
  <locale>on a park bench</locale>
  <activity>feeding birds</activity>
</setting>
```

Note If the **who** attribute is not supplied, the setting is assumed to be that of all participants in the language interaction.

<settingDesc> (setting description) describes the setting or settings within which a language interaction takes place, either as a prose description or as a series of setting elements.

Module corpus — *15. Language Corpora*

Attributes att.declarable (*@default*)

Used by model.profileDescPart

May contain

core: p

corpus: setting

linking: ab

Declaration

```
element settingDesc
{
   att.global.attributes,
   att.declarable.attributes,
   ( model.pLike+ | setting+ )
}
```

Example

```
<settingDesc>
  <p>Texts recorded in the Canadian Parliament building
     in Ottawa, between April and November 1988
  </p>
</settingDesc>
```

Note May contain a prose description organized as paragraphs, or a series of <setting> elements.

<settlement> contains the name of a settlement such as a city, town, or village identified as a single geo-political or administrative unit.

Module namesdates — *13. Names, Dates, People, and Places*

Attributes att.naming (*@key*, *@ref*, *@nymRef*) att.typed (*@type*, *@subtype*) att.datableatt.datable.w3c (*@period*, *@when*, *@notBefore*, *@notAfter*, *@from*, *@to*) att.datable.iso (*@when-iso*, *@notBefore-iso*, *@notAfter-iso*, *@from-iso*, *@to-iso*)

Used by model.placeNamePart

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address binaryObject cb choice corr date del distinct email emph expan foreign gap gloss graphic hi index lb measure measureGrp mentioned milestone name note num orig pb ptr ref reg rs sic soCalled term time title unclear

dictionaries: lang oRef oVar pRef pVar

figures: formula

gaiji: g

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident specDesc specList tag val

textcrit: app witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element settlement
{
   att.global.attributes,
   att.naming.attributes,
   att.typed.attributes,
   att.datable.w3c.attributes,
   att.datable.iso.attributes,
   macro.phraseSeq}
```

Example

```
<placeName>
  <settlement  type="town">Glasgow</settlement>
```

```
  <region>Scotland</region>
</placeName>
```

<sex> specifies the sex of a person.

Module namesdates — *13. Names, Dates, People, and Places*

Attributes att.editLike (*@cert*, *@resp*, *@evidence*, *@source*, *@precision*) att.datableatt.datable.w3c (*@period*, *@when*, *@notBefore*, *@notAfter*, *@from*, *@to*) att.datable.iso (*@when-iso*, *@notBefore-iso*, *@notAfter-iso*, *@from-iso*, *@to-iso*)

@value *Status* Optional

Datatype `data.sex`

Note Values for this attribute are taken from ISO 5218:1977 *Representation of Human Sexes*; 0 indicates unknown; 1 indicates male; 2 indicates female; and 9 indicates not applicable.

Used by model.persTraitLike

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address binaryObject cb choice corr date del distinct email emph expan foreign gap gloss graphic hi index lb measure measureGrp mentioned milestone name note num orig pb ptr ref reg rs sic soCalled term time title unclear

dictionaries: lang oRef oVar pRef pVar

figures: formula

gaiji: g

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident specDesc specList tag val

textcrit: app witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element sex
{
   att.global.attributes,
   att.editLike.attributes,
   att.datable.w3c.attributes,
   att.datable.iso.attributes,
```

```
   attribute value { data.sex }?,
   macro.phraseSeq}
```

Example

```
<sex value="2">female</sex>
```

<shift/> marks the point at which some paralinguistic feature of a series of utterances by any one speaker changes.

Module spoken — *8. Transcriptions of Speech*

Attributes att.ascribed (*@who*)

@feature a paralinguistic feature.

Status Optional

Legal values are: **tempo** speed of utterance.

loud loudness.

pitch pitch range.

tension tension or stress pattern.

rhythm rhythmic qualities.

voice voice quality.

@new specifies the new state of the paralinguistic feature specified.

Status Optional

Datatype `data.enumerated`

Values An open list (for an example of possible values, see *8.4.2. Synchronization and Overlap*)

Note If no value is specified, it is assumed that the feature concerned ceases to be remarkable. The value 'normal' has the same effect.

Used by model.global.spoken

May contain Empty element

Declaration

```
element shift
{
   att.global.attributes,
   att.ascribed.attributes,
   attribute feature
   {
      "tempo" | "loud" | "pitch" | "tension" | "rhythm" | "voice"
   }?,
   attribute new { data.enumerated }?,
   empty
}
```

Example

```
<u who="#LB">
  <shift feature="loud" new="f"/>Elizabeth
</u>
```

```
<u  who="#EB">Yes</u>
<u  who="#LB">
  <shift  feature="loud"/>Come and try this
<pause/>
  <shift  feature="loud"  new="ff"/>come on
</u>
<!-- ... -->
<listPerson  type="speakers">
  <person  xml:id="LB"/>
  <person  xml:id="EB"/>
</listPerson>
```

The word 'Elizabeth' is spoken loudly, the words 'Yes' and 'Come and try this' with normal volume, and the words 'come on' very loudly.

<sic> (latin for thus or so) contains text reproduced although apparently incorrect or inaccurate.

Module core — *3. Elements Available in All TEI Documents*

Attributes Global attributes only

Used by model.pPart.transcriptional model.choicePart

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address bibl biblStruct binaryObject cb choice cit corr date del desc distinct email emph expan foreign gap gloss graphic hi index label lb list listBibl measure measureGrp mentioned milestone name note num orig pb ptr q quote ref reg rs said sic soCalled stage term time title unclear

dictionaries: lang oRef oVar pRef pVar

drama: camera caption castList move sound tech view

figures: figure formula table

gaiji: g

header: biblFull

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material msDesc origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName listNym listOrg listPerson listPlace nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att classSpec code eg egXML elementSpec gi ident listRef macroSpec moduleRef moduleSpec specDesc specGrp specGrpRef specList tag val

textcrit: app listWit witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element sic { att.global.attributes, macro.paraContent }
```

Example

```
for his nose was as sharp as a pen, and <sic>a Table</sic> of green fields.
```

Source: [169]

Example If all that is desired is to call attention to the apparent problem in the copy text, <sic> may be used alone:

```
I don't know, Juan. It's so far in the past
now — how <sic>we can</sic> prove or disprove anyone's theories?
```

Example It is also possible, using the <choice> and <corr> elements, to provide a corrected reading:

```
I don't know, Juan. It's so far in the past
now — how <choice>
  <sic>we can</sic>
  <corr>can we</corr>
</choice> prove or disprove anyone's theories?
```

Example

```
for his nose was as sharp as a pen,
and <choice>
  <sic>a Table</sic>
  <corr>a' babbld</corr>
</choice> of green fields.
```

Source: [170]

<signatures> contains discussion of the leaf or quire signatures found within a codex.

Module msdescription — *10. Manuscript Description*

Attributes Global attributes only

Used by model.pPart.msdesc

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address binaryObject cb choice corr date del distinct email emph expan foreign gap gloss graphic hi index lb measure measureGrp mentioned milestone name note num orig pb ptr ref reg rs sic soCalled term time title unclear

dictionaries: lang oRef oVar pRef pVar

figures: formula

gaiji: g

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident specDesc specList tag val

textcrit: app witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element signatures { att.global.attributes, macro.phraseSeq }
```

Example

```
<signatures>Quire and leaf signatures in letters, [b]-v, and roman
numerals; those in quires 10 (1) and 17 (s) in red ink and different
from others; every third quire also signed with red crayon in arabic
numerals in the center lower margin of the first leaf recto: "2" for
quire 4 (f. 19), "3" for quire 7 (f. 43); "4," barely visible, for
quire 10 (f. 65), "5," in a later hand, for quire 13 (f. 89), "6," in
a later hand, for quire 16 (f. 113).</signatures>
```

<signed>

(signature) contains the closing salutation, etc., appended to a foreword, dedicatory epistle, or other division of a text.

Module textstructure — *4. Default Text Structure*

Attributes Global attributes only

Used by closer opener model.divBottomPart

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address binaryObject cb choice corr date del distinct email emph expan foreign gap gloss graphic hi index lb measure measureGrp mentioned milestone name note num orig pb ptr ref reg rs sic soCalled term time title unclear

dictionaries: lang oRef oVar pRef pVar

figures: formula

gaiji: g

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident specDesc specList tag val

textcrit: app witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element signed { att.global.attributes, macro.phraseSeq }
```

Example

```
<signed>Thine to command <name>Humph. Moseley</name>
</signed>
```

<soCalled> contains a word or phrase for which the author or narrator indicates a disclaiming of responsibility, for example by the use of scare quotes or italics.

Module core — *3. Elements Available in All TEI Documents*

Attributes Global attributes only

Used by model.emphLike

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address binaryObject cb choice corr date del distinct email emph expan foreign gap gloss graphic hi index lb measure measureGrp mentioned milestone name note num orig pb ptr ref reg rs sic soCalled term time title unclear

dictionaries: lang oRef oVar pRef pVar

figures: formula

gaiji: g

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident specDesc specList tag val

textcrit: app witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element soCalled { att.global.attributes, macro.phraseSeq }
```

Example

```
To edge his way along the crowded paths of life, warning
all human sympathy to keep its distance, was what the
knowing ones call <soCalled>nuts</soCalled> to Scrooge.
```

<socecStatus> (socio-economic status) contains an informal description of a person's perceived social or economic status.

Module namesdates — *13. Names, Dates, People, and Places*

Attributes att.datableatt.datable.w3c (*@period*, *@when*, *@notBefore*, *@notAfter*, *@from*, *@to*) att.datable.iso (*@when-iso*, *@notBefore-iso*, *@notAfter-iso*, *@from-iso*, *@to-iso*) att.editLike (*@cert*, *@resp*, *@evidence*, *@source*, *@precision*) att.naming (*@key*, *@ref*, *@nymRef*)

@scheme identifies the classification system or taxonomy in use.

Status Optional

Datatype `data.pointer`

Values Must identify a <taxonomy> element

@code identifies a status code defined within the classification system or taxonomy defined by the **source** attribute.

Status Optional

Datatype `data.pointer`

Values Must identify a <category> element

Used by model.persTraitLike

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address binaryObject cb choice corr date del distinct email emph expan foreign gap gloss graphic hi index lb measure measureGrp mentioned milestone name note num orig pb ptr ref reg rs sic soCalled term time title unclear

dictionaries: lang oRef oVar pRef pVar

figures: formula

gaiji: g

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident specDesc specList tag val

textcrit: app witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element socecStatus
{
   att.global.attributes,
   att.datable.w3c.attributes,
   att.datable.iso.attributes,
   att.editLike.attributes,
   att.naming.attributes,
   attribute scheme { data.pointer }?,
   attribute code { data.pointer }?,
   macro.phraseSeq}
```

Example

```
<socecStatus  scheme="#rg"  code="#ab1"/>
```

Example

```
<socecStatus>Status AB1 in the RG Classification scheme</socecStatus>
```

Note The content of this element may be used as an alternative to the more formal specification made possible by its attributes; it may also be used to supplement the formal specification with commentary or clarification.

<sound> describes a sound effect or musical sequence specified within a screen play or radio script.

Module drama — *7. Performance Texts*

Attributes In addition to global attributes

@type categorizes the sound in some respect, e.g. as music, special effect, etc.

Status Optional

Datatype `data.enumerated`

@discrete indicates whether the sound overlaps the surrounding speeches or interrupts them.

Status Optional

Datatype `data.xTruthValue`

Note The value **true** indicates that the sound is heard between the surrounding speeches; the value **false** indicates that the sound overlaps one or more of the surrounding speeches.

Used by model.stageLike

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address bibl biblStruct binaryObject cb choice cit corr date del desc distinct email emph expan foreign gap gloss graphic hi index label lb list listBibl measure measureGrp mentioned milestone name note num orig pb ptr q quote ref reg rs said sic soCalled stage term time title unclear

dictionaries: lang oRef oVar pRef pVar

drama: camera caption castList move sound tech view

figures: figure formula table

gaiji: g

header: biblFull

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material msDesc origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName listNym listOrg listPerson listPlace nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att classSpec code eg egXML elementSpec gi ident listRef macroSpec moduleRef moduleSpec specDesc specGrp specGrpRef specList tag val

textcrit: app listWit witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element sound
{
   att.global.attributes,
   attribute type { data.enumerated }?,
   attribute discrete { data.xTruthValue }?,
   macro.paraContent}
```

Example

```
<sp>
  <speaker>Benjy</speaker>
  <p>Now to business.</p>
</sp>
<sp>
  <speaker>Ford and Zaphod</speaker>
  <p>To business.</p>
</sp>
<sound  discrete="true">Glasses clink.</sound>
<sp>
  <speaker>Benjy</speaker>
  <p>I beg your pardon?</p>
</sp>
<sp>
  <speaker>Ford</speaker>
  <p>I'm sorry, I thought you were proposing a toast.</p>
</sp>
```

Note A specialized form of stage direction.

<source> describes the original source for the information contained with a manuscript description.

Module msdescription — *10. Manuscript Description*

Attributes Global attributes only

Used by recordHist

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address bibl biblStruct binaryObject cb choice cit corr date del desc distinct email emph expan foreign gap gloss graphic hi index l label lb lg list listBibl measure measureGrp mentioned milestone name note num orig p pb ptr q quote ref reg rs said sic soCalled sp stage term time title unclear

dictionaries: lang oRef oVar pRef pVar

drama: camera caption castList move sound tech view

figures: figure formula table

gaiji: g

header: biblFull

iso-fs: fLib fs fvLib

linking: ab alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material msDesc origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName listNym listOrg listPerson listPlace nameLink offset orgName persName placeName region roleName settlement state surname

nets: eTree forest forestGrp graph tree

spoken: incident kinesic pause shift u vocal writing

tagdocs: att classSpec code eg egXML elementSpec gi ident listRef macroSpec moduleRef moduleSpec schemaSpec specDesc specGrp specGrpRef specList tag val

textcrit: app listWit witDetail

textstructure: floatingText

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element source { att.global.attributes, macro.specialPara }
```

Example

```
<source>Derived from <ref>Stanley (1960)</ref>
</source>
```

<sourceDesc> (source description) supplies a description of the source text(s) from which an electronic text was derived or generated.

Module header — *2. The TEI Header*

Attributes att.declarable (*@default*)

Used by biblFull fileDesc

May contain

core: bibl biblStruct list listBibl p

header: biblFull

linking: ab

msdescription: msDesc

namesdates: listNym listOrg listPerson listPlace

spoken: recordingStmt scriptStmt

textcrit: listWit

Declaration

```
element sourceDesc
{
   att.global.attributes,
   att.declarable.attributes,
   (
      model.pLike+
    | ( model.biblLike | model.sourceDescPart | model.listLike )+
   )
}
```

Example

```
<sourceDesc>
  <p>No source: created in machine-readable form.</p>
</sourceDesc>
```

<sp> (speech) An individual speech in a performance text, or a passage presented as such in a prose or verse text.

Module core — *3. Elements Available in All TEI Documents*

Attributes att.ascribed (*@who*)

Used by model.divPart

May contain

analysis: interp interpGrp span spanGrp

certainty: certainty respons

core: cb cit gap index l lb lg milestone note p pb q quote said speaker stage

drama: camera caption move sound tech view

iso-fs: fLib fs fvLib

linking: ab alt altGrp anchor join joinGrp link linkGrp timeline

spoken: incident kinesic pause shift vocal writing

textcrit: witDetail

transcr: addSpan damageSpan delSpan fw space

Declaration

```
element sp
{
   att.global.attributes,
   att.ascribed.attributes,
```

```
   (
      model.global*,
      ( speaker, model.global* )?,
      (
         ( model.lLike | lg | model.pLike | model.stageLike | model.qLike ),
         model.global*
      )+
   )
}
```

Example

```
<sp>
  <speaker>The reverend Doctor Opimiam</speaker>
  <p>I do not think I have named a single unpresentable fish.</p>
</sp>
<sp>
  <speaker>Mr Gryll</speaker>
  <p>Bream, Doctor: there is not much to be said for bream.</p>
</sp>
<sp>
  <speaker>The Reverend Doctor Opimiam</speaker>
  <p>On the contrary, sir, I think there is much to be said for him.
     In the first place....</p>
  <p>Fish, Miss Gryll — I could discourse to you on fish by the hour:
     but for the present I will forbear...</p>
</sp>
```

Note The **who** attribute on this element may be used either in addition to the <speaker> element or as an alternative.

<space/> indicates the location of a significant space in the copy text.

Module transcr — *11. Representation of Primary Sources*

Attributes att.dimensions (*@unit*, *@quantity*, *@precision*, *@scope*)

@dim (dimension) indicates whether the space is horizontal or vertical.

Status Recommended

Legal values are: **horizontal** the space is horizontal.

vertical the space is vertical.

Note For irregular shapes in two dimensions, the value for this attribute should reflect the more important of the two dimensions. In conventional left-right scripts, a space with both vertical and horizontal components should be classed as **vertical**.

@resp (responsible party) indicates the individual responsible for identifying and measuring the space.

Status Optional

Datatype `data.pointer`

Values a pointer to one of the identifiers declared in the document header, associated with a person asserted as responsible for some aspect of the text's creation, transcription, editing, or encoding

Used by model.global.edit

May contain Empty element

Declaration

```
element space
{
   att.global.attributes,
   att.dimensions.attributes,
   attribute dim { "horizontal" | "vertical" }?,
   attribute resp { data.pointer }?,
   empty
}
```

Example

```
By god if wommen had writen storyes
As <space  quantity="7"  unit="minims"/> han within her oratoryes
```

Note This element should be used wherever it is desired to record an unusual space in the source text, e.g. space left for a word to be filled in later, for later rubrication, etc. It is not intended to be used to mark normal inter-word space or the like.

**** associates an interpretative annotation directly with a span of text.

Module analysis — *17. Simple Analytic Mechanisms*

Attributes att.interpLike (*@resp*, *@type*, *@inst*)

@from specifies the beginning of the passage being annotated; if not accompanied by a **to** attribute, then specifies the entire passage.

Status **Required**

Datatype `data.pointer`

Values The identifier of the element which occurs at the beginning of the passage.

@to specifies the end of the passage being annotated.

Status Optional

Datatype `data.pointer`

Values The identifier of the element which occurs at the end of the passage.

Used by spanGrp model.global.meta

May contain

analysis: interp interpGrp span spanGrp

certainty: certainty respons

core: abbr address cb choice date distinct email emph expan foreign gap gloss index lb measure measureGrp mentioned milestone name note num pb ptr ref rs soCalled term time title

dictionaries: lang

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident tag val

textcrit: witDetail

transcr: addSpan am damageSpan delSpan ex fw handShift space subst

Declaration

```
element span
{
   att.global.attributes,
   att.interpLike.attributes,
   attribute from { data.pointer },
   attribute to { data.pointer }?,
   macro.phraseSeq.limited}
```

Example

```
<p xml:id="para2">(The "aftermath" starts here)</p>
<p xml:id="para3">(The "aftermath" continues here)</p>
<p xml:id="para4">(The "aftermath" ends in this paragraph)</p>
<!-- ... -->
<span type="structure" from="#para2" to="#para4">aftermath</span>
```

<spanGrp> (span group) collects together span tags.

Module analysis — *17. Simple Analytic Mechanisms*

Attributes att.interpLike (*@resp*, *@type*, *@inst*)

Used by model.global.meta

May contain

analysis: span

Declaration

```
element spanGrp { att.global.attributes, att.interpLike.attributes, span* }
```

Example

```
<u xml:id="UU1">Can I have ten oranges and a kilo of bananas please?</u>
<u xml:id="UU2">Yes, anything else?</u>
<u xml:id="UU3">No thanks.</u>
<u xml:id="UU4">That'll be dollar forty.</u>
<u xml:id="UU5">Two dollars</u>
<u xml:id="UU6">Sixty, eighty, two dollars.
<anchor xml:id="UU6e"/>Thank you.<anchor xml:id="UU6f"/>
</u>
<spanGrp type="transactions">
  <span from="#UU1">sale request</span>
  <span from="#UU2" to="#UU3">sale compliance</span>
  <span from="#UU4">sale</span>
  <span from="#UU5" to="#UU6">purchase</span>
```

```
  <span from="#UU6e" to="#UU6f">purchase closure</span>
</spanGrp>
```

Source: [95]

<speaker> A specialized form of heading or label, giving the name of one or more speakers in a dramatic text or fragment.

Module core — *3. Elements Available in All TEI Documents*

Attributes Global attributes only

Used by sp

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address binaryObject cb choice corr date del distinct email emph expan foreign gap gloss graphic hi index lb measure measureGrp mentioned milestone name note num orig pb ptr ref reg rs sic soCalled term time title unclear

dictionaries: lang oRef oVar pRef pVar

figures: formula

gaiji: g

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident specDesc specList tag val

textcrit: app witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element speaker { att.global.attributes, macro.phraseSeq }
```

Example

```
<sp who="#ni #rsa">
  <speaker>Nancy and Robert</speaker>
  <stage type="delivery">(speaking simultaneously)</stage>
  <p>The future? ...</p>
</sp>
<list type="speakers">
  <item xml:id="ni"/>
```

```
  <item  xml:id="rsa"/>
</list>
```

Note This element is used to mark a speaker attribution as it appears in a dramatic text; the **who** attribute is used to point to another element which provides information about a speaker. Either or both may be used.

<specDesc/> (specification description) indicates that a description of the specified element or class should be included at this point within a document.

Module tagdocs — *22. Documentation Elements*

Attributes In addition to global attributes

@key (identifier) supplies the identifier of the documentary element or class for which a description is to be obtained.

Status Optional

Datatype `data.name`

```
<specDesc
   key="emph"/>
```

@atts (attributes) supplies attribute names for which descriptions should additionally be obtained.

Status Recommended

Datatype 0–∞ occurrences of `data.name` separated by whitespace

Values a whitespace-separated list of attribute names

```
<specDesc
   key="foreign"
   atts="usage xml:lang"/>
```

Note The attribute names listed may include both attributes inherited from a class and those defined explicitly for the associated element. If the **atts** attribute is not supplied, then descriptions for all non-inherited attributes are listed, along with references to any classes. If an empty string is supplied as the value for the **atts** attribute, then no description should be displayed.

Used by specList model.specDescLike

May contain Empty element

Declaration

```
element specDesc
{
   att.global.attributes,
   attribute key { data.name }?,
   attribute atts { list { data.name* } }?,
   empty
}
```

Example

```
<specDesc key="orth"/>
```

Note The description is usually displayed as a label and an item, with any list of values defined for the attribute as an embedded glossary list, No selection among the values is possible. The list of attributes may include some which are inherited by virtue of an element's class membership; descriptions for such attributes may also be retrieved using another <specDesc>, this time pointing at the relevant class.

<specGrp> (specification group) contains any convenient grouping of specifications for use within the current module.

Module tagdocs — *22. Documentation Elements*

Attributes Global attributes only

Used by model.oddDecl

May contain

core: l lg p sp

linking: ab

nets: eTree forest forestGrp graph tree

spoken: u

tagdocs: classSpec elementSpec listRef macroSpec moduleRef moduleSpec schemaSpec specGrp specGrpRef

textstructure: floatingText

Declaration

```
element specGrp
{
   att.global.attributes,
   ( model.oddDecl | model.oddRef | model.divPart )*
}
```

Example

```
<specGrp xml:id="xDAILC">
  <elementSpec ident="s">
<!-- ... -->
  </elementSpec>
  <elementSpec ident="cl">
<!-- ... -->
  </elementSpec>
  <elementSpec ident="w">
<!-- ... -->
  </elementSpec>
  <elementSpec ident="m">
<!-- ... -->
  </elementSpec>
  <elementSpec ident="c">
<!-- ... -->
  </elementSpec>
</specGrp>
```

This specification group with identifier xDAILC contains specifications for the elements <s>,<cl>,<w>, etc.

Note A specification group is referenced by means of its **xml:id** attribute. The declarations it contains may be included in a <module> element only by reference (using a <specGrpRef> element): it may not be nested within a <module> element. Different ODD processors may generate representations of the specifications contained by a <specGrp> in different concrete syntaxes. For P5 the intention is to generate modules using both XML and RELAX NG, and to use only the compressed RELAX NG syntax to represent them.

<specGrpRef/> (reference to a specification group) indicates that the declarations contained by the <specGrp> referenced should be inserted at this point.

Module tagdocs — *22. Documentation Elements*

Attributes In addition to global attributes

@target points at the specification group which logically belongs here.

Status **Required**

Datatype `data.pointer`

Used by schemaSpec model.oddRef

May contain Empty element

Declaration

```
element specGrpRef
{
   att.global.attributes,
   attribute target { data.pointer },
   empty
}
```

Example

```
<p>This part of
the module contains declarations for names of persons, places,
and organisations:
<specGrpRef target="#names.pers"/>
  <specGrpRef target="#names.place"/>
  <specGrpRef target="#names.org"/>
</p>
<!-- elsewhere -->
<specGrp xml:id="names.pers">
<!--... -->
</specGrp>
<!-- elsewhere -->
<specGrp xml:id="names.place">
<!--... -->
</specGrp>
<!-- elsewhere -->
<specGrp xml:id="names.org">
<!--... -->
</specGrp>
```

Note In ODD documentation processing, a <specGrpRef> usually produces a comment indicating that a set of declarations printed in another section will be inserted at this point in the <specGrp>

being discussed. In schema processing, the contents of the specified <specGrp> are made available for inclusion in the generated schema. The specification group identified by the **target** attribute will normally be part of the current ODD document.

<specList> (specification list) marks where a list of descriptions is to be inserted into the prose documentation.

Module tagdocs — *22. Documentation Elements*

Attributes Global attributes only

Used by model.specDescLike

May contain

tagdocs: specDesc

Declaration

```
element specList { att.global.attributes, specDesc+ }
```

Example

```
<specList>
  <specDesc key="lb" atts="ed"/>
  <specDesc key="div"/>
</specList>
```

<sponsor> specifies the name of a sponsoring organization or institution.

Module header — *2. The TEI Header*

Attributes Global attributes only

Used by titleStmt

May contain

analysis: interp interpGrp span spanGrp

certainty: certainty respons

core: abbr address cb choice date distinct email emph expan foreign gap gloss index lb measure measureGrp mentioned milestone name note num pb ptr ref rs soCalled term time title

dictionaries: lang

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident tag val

textcrit: witDetail

transcr: addSpan am damageSpan delSpan ex fw handShift space subst

Declaration

```
element sponsor { att.global.attributes, macro.phraseSeq.limited }
```

Example

```
<sponsor>Association for Computers and the Humanities</sponsor>
<sponsor>Association for Computational Linguistics</sponsor>
<sponsor>Association for Literary and Linguistic Computing</sponsor>
```

Note Sponsors give their intellectual authority to a project; they are to be distinguished from funders, who provide the funding but do not necessarily take intellectual responsibility.

<stage> (stage direction) contains any kind of stage direction within a dramatic text or fragment.

Module core — *3. Elements Available in All TEI Documents*

Attributes In addition to global attributes

@type indicates the kind of stage direction.

Status Recommended

Datatype `data.enumerated`

Suggested values include: **setting** describes a setting.

entrance describes an entrance.

exit describes an exit.

business describes stage business.

novelistic is a narrative, motivating stage direction.

delivery describes how a character speaks.

modifier gives some detail about a character.

location describes a location.

mixed more than one of the above

Used by model.stageLike

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address bibl biblStruct binaryObject cb choice cit corr date del desc distinct email emph expan foreign gap gloss graphic hi index l label lb lg list listBibl measure measureGrp mentioned milestone name note num orig p pb ptr q quote ref reg rs said sic soCalled sp stage term time title unclear

dictionaries: lang oRef oVar pRef pVar

drama: camera caption castList move sound tech view

figures: figure formula table

gaiji: g

header: biblFull

iso-fs: fLib fs fvLib

linking: ab alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material msDesc origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName listNym listOrg listPerson listPlace nameLink offset orgName persName placeName region roleName settlement state surname

nets: eTree forest forestGrp graph tree

spoken: incident kinesic pause shift u vocal writing

tagdocs: att classSpec code eg egXML elementSpec gi ident listRef macroSpec moduleRef moduleSpec schemaSpec specDesc specGrp specGrpRef specList tag val

textcrit: app listWit witDetail

textstructure: floatingText

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element stage
{
   att.global.attributes,
   attribute type
   {
      "setting"
    | "entrance"
    | "exit"
    | "business"
    | "novelistic"
    | "delivery"
    | "modifier"
    | "location"
    | "mixed"
    | xsd:Name
   }?,
   macro.specialPara}
```

Example

```
<stage type="setting">A curtain being drawn.</stage>
<stage type="setting">Music</stage>
<stage type="entrance">Enter Husband as being thrown off his
horse.</stage>
<stage type="exit">Exit pursued by a bear.</stage>
<stage type="business">He quickly takes the stone out.</stage>
<stage type="delivery">To Lussurioso.</stage>
<stage type="novelistic">Having had enough, and embarrassed for
the family.</stage>
<stage type="modifier">Disguised as Ansaldo.</stage>
<stage type="location">At a window.</stage>
<stage rend="inline" type="delivery">Aside.</stage>
```

<stamp> contains a word or phrase describing a stamp or similar device.

Module msdescription — *10. Manuscript Description*

Attributes att.typed (*@type*, *@subtype*) att.datableatt.datable.w3c (*@period*, *@when*, *@notBefore*, *@notAfter*, *@from*, *@to*) att.datable.iso (*@when-iso*, *@notBefore-iso*, *@notAfter-iso*, *@from-iso*, *@to-iso*)

Used by model.pPart.msdesc

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address binaryObject cb choice corr date del distinct email emph expan foreign gap gloss graphic hi index lb measure measureGrp mentioned milestone name note num orig pb ptr ref reg rs sic soCalled term time title unclear

dictionaries: lang oRef oVar pRef pVar

figures: formula

gaiji: g

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident specDesc specList tag val

textcrit: app witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element stamp
{
   att.global.attributes,
   att.typed.attributes,
   att.datable.w3c.attributes,
   att.datable.iso.attributes,
   macro.phraseSeq}
```

Example

```
<rubric>Apologyticu TTVLLIANI AC IGNORATIA IN XPO IHV<lb/>
SI NON LICET<lb/>
NOBIS RO<lb/>
manii imperii <stamp>Bodleian stamp</stamp>
  <lb/>
</rubric>
```

<state> contains a description of some status or quality attributed to a person, place, or organization at some specific time.

Module namesdates — *13. Names, Dates, People, and Places*

Attributes att.datableatt.datable.w3c (*@period*, *@when*, *@notBefore*, *@notAfter*, *@from*, *@to*) att.datable.iso (*@when-iso*, *@notBefore-iso*, *@notAfter-iso*, *@from-iso*, *@to-iso*) att.editLike (*@cert*, *@resp*, *@evidence*, *@source*, *@precision*) att.typed (*@type*, *@subtype*) att.naming (*@key*, *@ref*, *@nymRef*)

Used by state model.persStateLike model.orgStateLike model.placeStateLike

May contain

core: bibl biblStruct desc head label note p

header: biblFull

linking: ab

msdescription: msDesc

namesdates: state

textcrit: witDetail

Declaration

```
element state
{
   att.global.attributes,
   att.datable.w3c.attributes,
   att.datable.iso.attributes,
   att.editLike.attributes,
   att.typed.attributes,
   att.naming.attributes,
   (
      state+
    | ( model.headLike*, model.pLike+, ( model.noteLike | model.biblLike )* )
    | ( ( model.labelLike | model.noteLike | model.biblLike )* )
   )
}
```

Example

```
<person>
  <state  ref="#SCHOL"  type="status">
    <label>scholar</label>
  </state>
</person>
```

<stdVals>

(standard values) specifies the format used when standardized date or number values are supplied.

Module header — *2. The TEI Header*

Attributes att.declarable (*@default*)

Used by model.editorialDeclPart

May contain

core: p

linking: ab

Declaration

```
element stdVals
{
   att.global.attributes,
   att.declarable.attributes,
   model.pLike+
}
```

Example

```
<stdVals>
  <p>All integer numbers are left-filled with zeroes to 8 digits.</p>
</stdVals>
```

<street> a full street address including any name or number identifying a building as well as the name of the street or route on which it is located.

Module core — *3. Elements Available in All TEI Documents*

Attributes Global attributes only

Used by model.addrPart

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address binaryObject cb choice corr date del distinct email emph expan foreign gap gloss graphic hi index lb measure measureGrp mentioned milestone name note num orig pb ptr ref reg rs sic soCalled term time title unclear

dictionaries: lang oRef oVar pRef pVar

figures: formula

gaiji: g

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident specDesc specList tag val

textcrit: app witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element street { att.global.attributes, macro.phraseSeq }
```

Example

```
<street>via della Faggiola, 36</street>
```

Example

```
<street>
  <name>Duntaggin</name>, 110 Southmoor Road
</street>
```

Note The order and presentation of house names and numbers and street names, etc., may vary considerably in different countries. The encoding should reflect the order which is appropriate in the country concerned.

<stress> contains the stress pattern for a dictionary headword, if given separately.

Module dictionaries — *9. Dictionaries*

Attributes Global attributes only

Used by model.entryPart

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address bibl biblStruct binaryObject cb choice cit corr date del desc distinct email emph expan foreign gap gloss graphic hi index label lb list listBibl measure measureGrp mentioned milestone name note num orig pb ptr q quote ref reg rs said sic soCalled stage term time title unclear

dictionaries: lang oRef oVar pRef pVar

drama: camera caption castList move sound tech view

figures: figure formula table

gaiji: g

header: biblFull

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material msDesc origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName listNym listOrg listPerson listPlace nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att classSpec code eg egXML elementSpec gi ident listRef macroSpec moduleRef moduleSpec specDesc specGrp specGrpRef specList tag val

textcrit: app listWit witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element stress { att.global.attributes, macro.paraContent }
```

<string> (string value) represents the value part of a feature-value specification which contains a string.

Module iso-fs — *18. Feature Structures*

Attributes Global attributes only

Used by model.featureVal.single

May contain

gaiji: g

Declaration

```
element string { att.global.attributes, macro.xtext }
```

Example

```
<f  name="greeting">
  <string>Hello, world!</string>
</f>
```

<stringVal> contains the intended expansion for the entity documented by a <macroSpec> element, enclosed by quotation marks.

Module tagdocs — *22. Documentation Elements*

Attributes Global attributes only

Used by macroSpec

May contain Character data only

Declaration `element stringVal { att.global.attributes, text }`

Example

```
<stringVal>"the choice of quotes isn't always unimportant"</stringVal>
```

Example System entities should include the SYSTEM keyword within the content of this element, as shown:

```
<stringVal>SYSTEM 'teiclasses.ent"'</stringVal>
```

Note The content of this element is the replacement text for the named entity, including any keywords, and surrounded by appropriate quotation marks.

<subc> (subcategorization) contains subcategorization information (transitive/intransitive, countable/non-countable, etc.)

Module dictionaries — *9. Dictionaries*

Attributes att.lexicographic (*@expand*, *@norm*, *@split*, *@value*, *@orig*, *@location*, *@mergedIn*, *@opt*)

Used by model.entryPart model.gramPart

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address bibl biblStruct binaryObject cb choice cit corr date del desc distinct email emph expan foreign gap gloss graphic hi index label lb list listBibl measure measureGrp mentioned milestone name note num orig pb ptr q quote ref reg rs said sic soCalled stage term time title unclear

dictionaries: lang oRef oVar pRef pVar

drama: camera caption castList move sound tech view

figures: figure formula table

gaiji: g

header: biblFull

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material msDesc origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName listNym listOrg listPerson listPlace nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att classSpec code eg egXML elementSpec gi ident listRef macroSpec moduleRef moduleSpec specDesc specGrp specGrpRef specList tag val

textcrit: app listWit witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element subc
{
   att.global.attributes,
   att.lexicographic.attributes,
   macro.paraContent}
```

Example

```
<entry>
  <form>
    <orth>médire</orth>
  </form>
  <gramGrp>
    <subc>t ind</subc>
  </gramGrp>
</entry>
```

<subst> (substitution) groups one or more deletions with one or more additions when the combination is to be regarded as a single intervention in the text.

Module transcr — *11. Representation of Primary Sources*

Attributes att.transcriptional (*@hand*, *@status*, *@seq*) att.editLike (*@cert*, *@resp*, *@evidence*, *@source*, *@precision*)

Used by model.pPart.editorial

May contain

core: add corr del orig reg sic unclear

textcrit: app

transcr: damage restore supplied

Declaration

```
element subst
{
   att.global.attributes,
   att.transcriptional.attributes,
   att.editLike.attributes,
   ( ( model.pPart.transcriptional ), model.pPart.transcriptional+ )
}
```

Example

```
... are all included. <del  hand="#RG">It is</del>
<subst>
  <add>T</add>
  <del>t</del>
</subst>he expressed
```

Note Although a substitution may contain any mixture of additions and deletions; there should be an addition for each deletion bearing the same sequence number. This constraint cannot be modelled in the schema language currently deployed.

<summary> contains a brief summary of the intellectual content of an item, provided by the cataloguer.

Module msdescription — *10. Manuscript Description*

Attributes Global attributes only

Used by handDesc history msContents

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address binaryObject cb choice corr date del distinct email emph expan foreign gap gloss graphic hi index lb measure measureGrp mentioned milestone name note num orig pb ptr ref reg rs sic soCalled term time title unclear

dictionaries: lang oRef oVar pRef pVar

figures: formula

gaiji: g

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident specDesc specList tag val

textcrit: app witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element summary { att.global.attributes, macro.phraseSeq }
```

Example

```
<summary>This item consists of three books with a prologue and an epilogue.
</summary>
```

<superEntry> groups successive entries for a set of homographs.

Module dictionaries — *9. Dictionaries*

Attributes att.entryLike (*@type*, *@sortKey*)

Used by model.entryLike model.entryPart

May contain

dictionaries: dictScrap entry form

Declaration

```
element superEntry
{
   att.global.attributes,
   att.entryLike.attributes,
   ( ( form?, entry+ ) | dictScrap )
}
```

Example

```
<superEntry>
  <form>
    <orth>abandon</orth>
    <hyph>a|ban|don</hyph>
    <pron>@"band@n</pron>
  </form>
  <entry  n="1">
    <gramGrp>
      <pos>v</pos>
      <subc>T1</subc>
```

```
    </gramGrp>
    <sense n="1">
      <def>to leave completely and for ever ... </def>
    </sense>
    <sense n="2"/>
  </entry>
  <entry n="2">
    <gramGrp>
      <pos>n</pos>
      <subc>U</subc>
    </gramGrp>
    <def>the state when one's feelings and actions are uncontrolled; freedom from
        control</def>
  </entry>
</superEntry>
```

<supplied> signifies text supplied by the transcriber or editor for any reason, typically because the original cannot be read because of physical damage or loss to the original.

Module transcr — *11. Representation of Primary Sources*

Attributes att.editLike (*@cert*, *@resp*, *@evidence*, *@source*, *@precision*)

@reason indicates why the text has had to be supplied.

Status Optional

Datatype 1–∞ occurrences of `data.word` separated by whitespace

Values any phrase describing the difficulty, e.g. *overbinding*, *faded ink*, *lost folio*, *omitted in original*.

Used by model.pPart.transcriptional

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address bibl biblStruct binaryObject cb choice cit corr date del desc distinct email emph expan foreign gap gloss graphic hi index label lb list listBibl measure measureGrp mentioned milestone name note num orig pb ptr q quote ref reg rs said sic soCalled stage term time title unclear

dictionaries: lang oRef oVar pRef pVar

drama: camera caption castList move sound tech view

figures: figure formula table

gaiji: g

header: biblFull

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material msDesc origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName listNym listOrg listPerson listPlace nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att classSpec code eg egXML elementSpec gi ident listRef macroSpec moduleRef moduleSpec specDesc specGrp specGrpRef specList tag val

textcrit: app listWit witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element supplied
{
   att.global.attributes,
   att.editLike.attributes,
   attribute reason { list { data.word+ } }?,
   macro.paraContent}
```

Example

```
I am dr Sr yr
<supplied reason="illegible" source="amanuensis copy">very humble Servt</supplied>
Sydney Smith
```

Note The <damage>, <gap>, <del>, <unclear> and <supplied> elements may be closely allied in use. See section *11.5.2. Use of the <gap>, <del>, <damage>, <unclear>, and <supplied> Elements in Combination* for discussion of which element is appropriate for which circumstance.

<support> contains a description of the materials etc. which make up the physical support for the written part of a manuscript.

Module msdescription — *10. Manuscript Description*

Attributes Global attributes only

Used by supportDesc

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address bibl biblStruct binaryObject cb choice cit corr date del desc distinct email emph expan foreign gap gloss graphic hi index l label lb lg list listBibl measure measureGrp mentioned milestone name note num orig p pb ptr q quote ref reg rs said sic soCalled sp stage term time title unclear

dictionaries: lang oRef oVar pRef pVar

drama: camera caption castList move sound tech view

figures: figure formula table

gaiji: g

header: biblFull

iso-fs: fLib fs fvLib

linking: ab alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material msDesc origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName listNym listOrg listPerson listPlace nameLink offset orgName persName placeName region roleName settlement state surname

nets: eTree forest forestGrp graph tree

spoken: incident kinesic pause shift u vocal writing

tagdocs: att classSpec code eg egXML elementSpec gi ident listRef macroSpec moduleRef moduleSpec schemaSpec specDesc specGrp specGrpRef specList tag val

textcrit: app listWit witDetail

textstructure: floatingText

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element support { att.global.attributes, macro.specialPara }
```

Example

```
<objectDesc  form="roll">
  <supportDesc>
    <support> Parchment roll with <material>silk</material> ribbons.
    </support>
  </supportDesc>
</objectDesc>
```

<supportDesc> (support description) groups elements describing the physical support for the written part of a manuscript.

Module msdescription — *10. Manuscript Description*

Attributes In addition to global attributes

@material a short project-defined name for the material composing the majority of the support

Status Optional

Datatype `data.enumerated`

Values are: **paper**

parch (parchment)

mixed

Used by objectDesc

May contain

core: p

header: extent

linking: ab

msdescription: collation condition foliation support

Declaration

```
element supportDesc
{
   att.global.attributes,
   attribute material { data.enumerated }?,
   ( model.pLike+ | ( support?, extent?, foliation*, collation?, condition? ) )
}
```

Example

```
<supportDesc>
  <support> Parchment roll with <material>silk</material> ribbons.
  </support>
</supportDesc>
```

<surface> defines a written surface in terms of a rectangular coordinate space, optionally grouping one or more graphic representations of that space, and rectangular zones of interest within it.

Module transcr — *11. Representation of Primary Sources*

Attributes att.coordinated (*@ulx*, *@uly*, *@lrx*, *@lry*) att.declaring (*@decls*)

@start points to an element which encodes the starting position of the text corresponding to the inscribed part of the surface.

Status Optional

Datatype `data.pointer`

Used by facsimile

May contain

core: binaryObject desc gloss graphic

figures: formula

tagdocs: altIdent equiv

transcr: zone

Declaration

```
element surface
{
   att.global.attributes,
   att.coordinated.attributes,
   att.declaring.attributes,
   attribute start { data.pointer }?,
   ( model.glossLike*, model.graphicLike*, zone* )
}
```

Example

```
<facsimile>
  <surface
     ulx="0"
     uly="0"
     lrx="200"
     lry="300">
    <graphic  url="Bovelles-49r.png"/>
```

```
  </surface>
</facsimile>
```

Note The <surface> element represents a rectangular area of any physical surface forming part of the source material. This may be a sheet of paper, one face of a monument, a billboard, a papyrus scroll, or indeed any 2-dimensional surface.The coordinate space defined by this element may be thought of as a grid **lrx** - **ulx** units wide and **uly** - **lry** units high. This grid is superimposed on the whole of any image directly contained by the <surface> element. The coordinate values used by every <zone> element contained by this surface are to be understood with reference to the same grid.

<surname> contains a family (inherited) name, as opposed to a given, baptismal, or nick name.

Module namesdates — *13. Names, Dates, People, and Places*

Attributes att.personal (*@full*, *@sort*) att.naming (*@key*, *@ref*, *@nymRef*) att.typed (*@type*, *@subtype*)

Used by model.persNamePart

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address binaryObject cb choice corr date del distinct email emph expan foreign gap gloss graphic hi index lb measure measureGrp mentioned milestone name note num orig pb ptr ref reg rs sic soCalled term time title unclear

dictionaries: lang oRef oVar pRef pVar

figures: formula

gaiji: g

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident specDesc specList tag val

textcrit: app witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element surname
{
   att.global.attributes,
   att.personal.attributes,
   att.naming.attributes,
```

```
  att.typed.attributes,
  macro.phraseSeq}
```

Example

```
<surname type="combine">St John Stevas</surname>
```

<surrogates> contains information about any digital or photographic representations of the manuscript being described which may exist in the holding institution or elsewhere.

Module msdescription — *10. Manuscript Description*

Attributes Global attributes only

Used by additional

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address bibl biblStruct binaryObject cb choice cit corr date del desc distinct email emph expan foreign gap gloss graphic hi index l label lb lg list listBibl measure measureGrp mentioned milestone name note num orig p pb ptr q quote ref reg rs said sic soCalled sp stage term time title unclear

dictionaries: lang oRef oVar pRef pVar

drama: camera caption castList move sound tech view

figures: figure formula table

gaiji: g

header: biblFull

iso-fs: fLib fs fvLib

linking: ab alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material msDesc origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName listNym listOrg listPerson listPlace nameLink offset orgName persName placeName region roleName settlement state surname

nets: eTree forest forestGrp graph tree

spoken: incident kinesic pause shift u vocal writing

tagdocs: att classSpec code eg egXML elementSpec gi ident listRef macroSpec moduleRef moduleSpec schemaSpec specDesc specGrp specGrpRef specList tag val

textcrit: app listWit witDetail

textstructure: floatingText

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element surrogates { att.global.attributes, macro.specialPara }
```

Example

```
<surrogates>
  <p>
    <bibl>
      <title  type="gmd">diapositive</title>
      <idno>AM 74 a, fol.</idno>
      <date>May 1984</date>
    </bibl>
    <bibl>
      <title  type="gmd">b/w prints</title>
      <idno>AM 75 a, fol.</idno>
      <date>1972</date>
    </bibl>
  </p>
</surrogates>
```

<syll> (syllabification) contains the syllabification of the headword.

Module dictionaries — *9. Dictionaries*

Attributes att.lexicographic (*@expand*, *@norm*, *@split*, *@value*, *@orig*, *@location*, *@mergedIn*, *@opt*)

Used by model.entryPart model.formPart

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address bibl biblStruct binaryObject cb choice cit corr date del desc distinct email emph expan foreign gap gloss graphic hi index label lb list listBibl measure measureGrp mentioned milestone name note num orig pb ptr q quote ref reg rs said sic soCalled stage term time title unclear

dictionaries: lang oRef oVar pRef pVar

drama: camera caption castList move sound tech view

figures: figure formula table

gaiji: g

header: biblFull

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material msDesc origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName listNym listOrg listPerson listPlace nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att classSpec code eg egXML elementSpec gi ident listRef macroSpec moduleRef moduleSpec specDesc specGrp specGrpRef specList tag val

textcrit: app listWit witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element syll
{
   att.global.attributes,
   att.lexicographic.attributes,
   macro.paraContent}
```

Example

```
<form>
  <orth>area</orth>
  <hyph>ar|ea</hyph>
  <syll>ar|e|a</syll>
</form>
```

<symbol/> (symbolic value) represents the value part of a feature-value specification which contains one of a finite list of symbols.

Module iso-fs — *18. Feature Structures*

Attributes In addition to global attributes

@value supplies the symbolic value for the feature, one of a finite list that may be specified in a feature declaration.

Status **Required**

Datatype `data.word`

Values A string, e.g. feminine.

Used by model.featureVal.single

May contain Empty element

Declaration

```
element symbol { att.global.attributes, attribute value { data.word }, empty }
```

Example

```
<f  name="gender">
  <symbol  value="feminine"/>
</f>
```

<table> contains text displayed in tabular form, in rows and columns.

Module figures — *14. Tables, Formulæ, and Graphics*

Attributes In addition to global attributes

@rows indicates the number of rows in the table.

Status Optional

Datatype `data.count`

Values If no number is supplied, an application must calculate the number of rows.

Note Rows should be presented from top to bottom.

@cols (columns) indicates the number of columns in each row of the table.

Status Optional

Datatype `data.count`

Values If no number is supplied, an application must calculate the number of columns.

Note Within each row, columns should be presented left to right.

Used by model.inter

May contain

analysis: interp interpGrp span spanGrp

certainty: certainty respons

core: cb gap head index lb milestone note pb

figures: row

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp timeline

spoken: incident kinesic pause shift vocal writing

textcrit: witDetail

transcr: addSpan damageSpan delSpan fw space

Declaration

```
element table
{
   att.global.attributes,
   attribute rows { data.count }?,
   attribute cols { data.count }?,
   ( ( model.headLike | model.global )*, ( row, model.global* )+ )
}
```

Example

```
<table rows="4" cols="4">
  <head>Poor Men's Lodgings in Norfolk (Mayhew, 1843)</head>
  <row role="label">
    <cell role="data"/>
    <cell role="data">Dossing Cribs or Lodging Houses</cell>
    <cell role="data">Beds</cell>
    <cell role="data">Needys or Nightly Lodgers</cell>
  </row>
  <row role="data">
    <cell role="label">Bury St Edmund's</cell>
    <cell role="data">5</cell>
    <cell role="data">8</cell>
    <cell role="data">128</cell>
  </row>
  <row role="data">
    <cell role="label">Thetford</cell>
    <cell role="data">3</cell>
    <cell role="data">6</cell>
    <cell role="data">36</cell>
  </row>
  <row role="data">
    <cell role="label">Attleboro'</cell>
```

```
      <cell role="data">3</cell>
      <cell role="data">5</cell>
      <cell role="data">20</cell>
    </row>
    <row role="data">
      <cell role="label">Wymondham</cell>
      <cell role="data">1</cell>
      <cell role="data">11</cell>
      <cell role="data">22</cell>
    </row>
  </table>
```

Note Contains an optional heading and a series of rows.Any rendition information should be supplied using the global **rend** attribute, at the table, row, or cell level as appropriate.

<tag> contains text of a complete start- or end-tag, possibly including attribute specifications, but excluding the opening and closing markup delimiter characters.

Module tagdocs — *22. Documentation Elements*

Attributes In addition to global attributes

@type indicates the type of XML tag intended

Status Optional

Legal values are: **start** a start-tag, with delimiters < and > is intended

end an end-tag, with delimiters </ and > is intended

empty a empty tag, with delimiters < and /> is intended

pi a pi (processing instruction), with delimiters <? and ?> is intended

comment a comment, with delimiters <!-- and --> is intended

ms a marked-section, with delimiters <[CDATA[and]]> is intended

@scheme supplies the name of the schema in which this tag is defined.

Status Optional

Legal values are: **TEI** (text encoding initiative) This tag is defined as part of the TEI scheme. **[Default]**

DBK (docbook) this tag is part of the Docbook scheme.

XX (unknown) this tag is part of an unknown scheme.

Used by model.phrase.xml

May contain Character data only

Declaration

```
element tag
{
   att.global.attributes,
   attribute type { "start" | "end" | "empty" | "pi" | "comment" | "ms" }?,
   attribute scheme { "TEI" | "DBK" | "XX" }?,
   text
}
```

Example

```
Mark the start of each italicised phrase with a <tag>hi rend="it"</tag>
tag, and its end with a <tag  type="end">hi</tag> tag.

<tag  type="comment">Example updated on 2008-04-05</tag>
```

<tagUsage> supplies information about the usage of a specific element within a text.

Module header — *2. The TEI Header*

Attributes In addition to global attributes

@gi (element name) the name (generic identifier) of the element indicated by the tag.
Status **Required**
Datatype `data.name`
Values the name of an element within the namespace indicated by the parent <namespace> element

@occurs specifies the number of occurrences of this element within the text.
Status Recommended
Datatype `data.count`
Values an integer number greater than zero

@withId (with unique identifier) specifies the number of occurrences of this element within the text which bear a distinct value for the global **xml:id** attribute.
Status Recommended
Datatype `data.count`
Values an integer number greater than zero

@render specifies the identifier of a <rendition> element which defines how this element is to be rendered.
Status Optional
Datatype `data.pointer`
Values an identifier specified as the value of the **xml:id** attribute on some <rendition> element in the current document.

Used by namespace

May contain

core: abbr address bibl biblStruct choice cit date desc distinct email emph expan foreign gloss label list listBibl measure measureGrp mentioned name num ptr q quote ref rs said soCalled stage term time title

dictionaries: lang

drama: camera caption castList move sound tech view

figures: figure table

header: biblFull

msdescription: catchwords depth dimensions height heraldry locus material msDesc origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName listNym listOrg listPerson listPlace nameLink offset orgName persName placeName region roleName settlement state surname

tagdocs: att classSpec code eg egXML elementSpec gi ident listRef macroSpec moduleRef moduleSpec specGrp specGrpRef tag val

textcrit: listWit

transcr: am ex handShift subst

Declaration

```
element tagUsage
{
   att.global.attributes,
   attribute gi { data.name },
   attribute occurs { data.count }?,
   attribute withId { data.count }?,
   attribute render { data.pointer }?,
   macro.limitedContent}
```

Example

```
<tagsDecl>
  <rendition xml:id="it">Render using a slant or italic variant
    on the current font</rendition>
<!-- ... -->
  <namespace name="http://www.tei-c.org/ns/1.0">
    <tagUsage
       gi="hi"
       occurs="28"
       withId="2"
       render="#it">Used to mark English words italicised in the copy text.</tagUsage>
    <tagUsage gi="foreign" render="#it">Used to mark non-English
        words in the copy text.</tagUsage>
<!-- ... -->
  </namespace>
</tagsDecl>
```

<tagsDecl> (tagging declaration) provides detailed information about the tagging applied to a document.

Module header — *2. The TEI Header*

Attributes Global attributes only

Used by model.encodingPart

May contain

header: namespace rendition

Declaration

```
element tagsDecl { att.global.attributes, ( rendition*, namespace* ) }
```

Example

```
<tagsDecl>
  <rendition xml:id="rend-it">to be rendered in italic font</rendition>
  <namespace name="http://www.tei-c.org/ns/1.0">
    <tagUsage gi="hi" occurs="467" render="#rend-it"/>
    <tagUsage gi="title" occurs="45" render="#rend-it"/>
```

```
  </namespace>
  <namespace name="http://docbook.org/ns/docbook">
    <tagUsage gi="para" occurs="10"/>
  </namespace>
</tagsDecl>
```

<taxonomy> defines a typology used to classify texts either implicitly, by means of a bibliographic citation, or explicitly by a structured taxonomy.

Module header — *2. The TEI Header*

Attributes Global attributes only

Used by classDecl

May contain

core: bibl biblStruct desc gloss

header: biblFull category

msdescription: msDesc

tagdocs: altIdent equiv

Declaration

```
element taxonomy
{
   att.global.attributes,
   ( model.glossLike* | category+ | ( ( model.biblLike ), category* ) )
}
```

Example

```
<taxonomy xml:id="tax.b">
  <bibl>Brown Corpus</bibl>
  <category xml:id="tax.b.a">
    <catDesc>Press Reportage</catDesc>
    <category xml:id="tax.b.a1">
      <catDesc>Daily</catDesc>
    </category>
    <category xml:id="tax.b.a2">
      <catDesc>Sunday</catDesc>
    </category>
    <category xml:id="tax.b.a3">
      <catDesc>National</catDesc>
    </category>
    <category xml:id="tax.b.a4">
      <catDesc>Provincial</catDesc>
    </category>
    <category xml:id="tax.b.a5">
      <catDesc>Political</catDesc>
    </category>
    <category xml:id="tax.b.a6">
      <catDesc>Sports</catDesc>
    </category>
  </category>
  <category xml:id="tax.b.d">
```

```
      <catDesc>Religion</catDesc>
      <category xml:id="tax.b.d1">
        <catDesc>Books</catDesc>
      </category>
      <category xml:id="tax.b.d2">
        <catDesc>Periodicals and tracts</catDesc>
      </category>
    </category>
  </taxonomy>
```

<tech> (technical stage direction) describes a special-purpose stage direction that is not meant for the actors.

Module drama — *7. Performance Texts*

Attributes In addition to global attributes

@type categorizes the technical stage direction.

Status Optional

Legal values are: **light** a lighting cue

sound a sound cue

prop a prop cue

block a blocking instruction

@perf (performance) identifies the performance or performances to which this technical direction applies.

Status Optional

Datatype `data.enumerated`

Values The IDREFS are derived from the **xml:id** attribute on a <performance> element.

Used by model.stageLike

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address bibl biblStruct binaryObject cb choice cit corr date del desc distinct email emph expan foreign gap gloss graphic hi index label lb list listBibl measure measureGrp mentioned milestone name note num orig pb ptr q quote ref reg rs said sic soCalled stage term time title unclear

dictionaries: lang oRef oVar pRef pVar

drama: camera caption castList move sound tech view

figures: figure formula table

gaiji: g

header: biblFull

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material msDesc origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName listNym listOrg listPerson listPlace nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att classSpec code eg egXML elementSpec gi ident listRef macroSpec moduleRef moduleSpec specDesc specGrp specGrpRef specList tag val

textcrit: app listWit witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element tech
{
   att.global.attributes,
   attribute type { "light" | "sound" | "prop" | "block" }?,
   attribute perf { data.enumerated }?,
   macro.paraContent}
```

Example

```
<tech type="light">Red spot on his face</tech>
```

<teiCorpus> contains the whole of a TEI encoded corpus, comprising a single corpus header and one or more TEI elements, each containing a single text header and a text.

Module core — *3. Elements Available in All TEI Documents*

Attributes In addition to global attributes

@version The version of the TEI scheme

Status Optional

Datatype `xsd:decimal`

Values A number identifying the version of the TEI guidelines

Used by teiCorpus

May contain

core: teiCorpus

header: teiHeader

textstructure: TEI

Declaration

```
element teiCorpus
{
   att.global.attributes,
   attribute version { xsd:decimal }?,
   ( teiHeader, ( TEI | teiCorpus )+ )
}
```

Example

```
<teiCorpus>
  <teiHeader>
<!-- header for corpus -->
  </teiHeader>
  <TEI>
    <teiHeader>
<!-- header for first text -->
    </teiHeader>
    <text>
<!-- content of first text -->
    </text>
  </TEI>
  <TEI>
    <teiHeader>
<!-- header for second text -->
    </teiHeader>
    <text>
<!-- content of second text -->
    </text>
  </TEI>
<!-- more TEI elements here -->
</teiCorpus>
```

Note Must contain one TEI header for the corpus, and a series of <TEI> elements, one for each text.This element is mandatory when applicable.

<teiHeader> (TEI Header) supplies the descriptive and declarative information making up an electronic title page prefixed to every TEI-conformant text.

Module header — *2. The TEI Header*

Attributes In addition to global attributes

@type specifies the kind of document to which the header is attached, for example whether it is a corpus or individual text.

Status Optional

Datatype `data.enumerated`

Sample values include: **text** the header is attached to a single text. **[Default]**

corpus the header is attached to a corpus.

Used by TEI teiCorpus

May contain

header: encodingDesc fileDesc profileDesc revisionDesc

Declaration

```
element teiHeader
{
   att.global.attributes,
   attribute type { data.enumerated }?,
   ( fileDesc, model.headerPart*, revisionDesc? )
}
```

Example

```
<teiHeader>
  <fileDesc>
    <titleStmt>
      <title>Shakespeare: the first folio (1623) in electronic form</title>
      <author>Shakespeare, William (1564–1616)</author>
      <respStmt>
        <resp>Originally prepared by</resp>
        <name>Trevor Howard-Hill</name>
      </respStmt>
      <respStmt>
        <resp>Revised and edited by</resp>
        <name>Christine Avern-Carr</name>
      </respStmt>
    </titleStmt>
    <publicationStmt>
      <distributor>Oxford Text Archive</distributor>
      <address>
        <addrLine>13 Banbury Road, Oxford OX2 6NN, UK</addrLine>
      </address>
      <idno  type="OTA">119</idno>
      <availability>
        <p>Freely available on a non-commercial basis.</p>
      </availability>
      <date  when="1968">1968</date>
    </publicationStmt>
    <sourceDesc>
      <bibl>The first folio of Shakespeare, prepared by Charlton Hinman
            (The Norton Facsimile, 1968)</bibl>
    </sourceDesc>
  </fileDesc>
  <encodingDesc>
    <projectDesc>
      <p>Originally prepared for use in the production of a series of
            old-spelling concordances in 1968, this text was extensively
            checked and revised for use during the editing of the new Oxford
            Shakespeare (Wells and Taylor, 1989).</p>
    </projectDesc>
    <editorialDecl>
      <correction>
        <p>Turned letters are silently corrected.</p>
      </correction>
      <normalization>
        <p>Original spelling and typography is retained, except
                that long s and ligatured forms are not encoded.</p>
      </normalization>
    </editorialDecl>
    <refsDecl  xml:id="ASLREF">
      <cRefPattern
         matchPattern="(\S+) ([^.]+)\.(.*)"
         replacementPattern="#xpath(//div1[@n='$1']/div2/[@n='$2']//lb[@n='$3'])">
        <p>A reference is created by assembling the following,
                in the reverse order as that listed here:
        <list>
           <item>the <att>n</att> value of the preceding <gi>lb</gi>
           </item>
           <item>a period</item>
           <item>the <att>n</att> value of the ancestor <gi>div2</gi>
```

```
            </item>
            <item>a space</item>
            <item>the <att>n</att> value of the parent <gi>div1</gi>
            </item>
          </list>
        </p>
      </cRefPattern>
    </refsDecl>
  </encodingDesc>
  <revisionDesc>
    <list>
      <item>
        <date  when="1989-04-12">12 Apr 89</date> Last checked by CAC</item>
      <item>
        <date  when="1989-03-01">1 Mar 89</date> LB made new file</item>
    </list>
  </revisionDesc>
</teiHeader>
```

Note One of the few elements unconditionally required in any TEI document.

<term> contains a single-word, multi-word, or symbolic designation which is regarded as a technical term.

Module core — *3. Elements Available in All TEI Documents*

Attributes att.declaring (*@decls*) att.typed (*@type*, *@subtype*)

@target identifies the associated <gloss> element by an absolute or relative URI reference

Status Optional

Datatype `data.pointer`

Values should be a valid URI reference that resolves to a <gloss> element

@cRef identifies the associated <gloss> element using a canonical reference from a scheme defined in a <refsDecl> element in the TEI header

Status Optional

Datatype `data.pointer`

Values the result of applying the algorithm for the resolution of canonical references (described in section *16.2.5. Canonical References*) should be a valid URI reference that resolves to a <gloss> element

Note The <refsDecl> to use may be indicated with the **decls** attribute.

@sortKey supplies the sort key for this term in an index.

Status Optional

Datatype `data.word`

Values any string of Unicode characters.

```
David's other principal backer, Josiah ha-Kohen
<index
   indexName="NAMES">
  <term
     sortKey="Azarya_Josiah_Kohen">Josiah ha-Kohen b.  Azarya</term>
```

```
</index> b. Azarya, son of one of the last gaons of Sura
was David's own first cousin.
```

Note The sort key is used to determine the sequence and grouping of entries in an index; if this attribute is not supplied, the textual content of the element is used for this purpose.

Used by index keywords model.emphLike

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address binaryObject cb choice corr date del distinct email emph expan foreign gap gloss graphic hi index lb measure measureGrp mentioned milestone name note num orig pb ptr ref reg rs sic soCalled term time title unclear

dictionaries: lang oRef oVar pRef pVar

figures: formula

gaiji: g

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident specDesc specList tag val

textcrit: app witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element term
{
   att.global.attributes,
   att.declaring.attributes,
   att.typed.attributes,
   ( attribute target { data.pointer }? | attribute cRef { data.pointer }? ),
   attribute sortKey { data.word }?,
   macro.phraseSeq}
```

Example

```
A computational device that infers structure from grammatical
strings of words is known as a <term>parser</term>, and much
of the history of NLP over the last 20 years has been occupied
with the design of parsers.
```

Example

```
We may define <term  xml:id="TDPV"  rend="sc">discoursal point of view</term>
as
<gloss  target="#TDPV">the relationship, expressed through discourse
structure, between the implied author or some other addresser,
and the fiction.</gloss>
```

Note This element is used to supply the form under which an index entry is to be made for the location of a parent <index> element. In formal terminological work, there is frequently discussion over whether terms must be atomic or may include multi-word lexical items, symbolic designations, or phraseological units. The <term> element may be used to mark any of these. No position is taken on the philosophical issue of what a term can be; the looser definition simply allows the <term> element to be used by practitioners of any persuasion.The **target** and **cRef** attributes are mutually exclusive.

<terrain> contains information about the physical terrain of a place.

Module namesdates — *13. Names, Dates, People, and Places*

Attributes att.datableatt.datable.w3c (*@period*, *@when*, *@notBefore*, *@notAfter*, *@from*, *@to*) att.datable.iso (*@when-iso*, *@notBefore-iso*, *@notAfter-iso*, *@from-iso*, *@to-iso*) att.editLike (*@cert*, *@resp*, *@evidence*, *@source*, *@precision*) att.naming (*@key*, *@ref*, *@nymRef*) att.typed (*@type*, *@subtype*)

Used by terrain model.placeTraitLike

May contain

core: bibl biblStruct desc head label note p

header: biblFull

linking: ab

msdescription: msDesc

namesdates: terrain

textcrit: witDetail

Declaration

```
element terrain
{
   att.global.attributes,
   att.datable.w3c.attributes,
   att.datable.iso.attributes,
   att.editLike.attributes,
   att.naming.attributes,
   att.typed.attributes,
   (
      model.headLike*,
      ( ( model.pLike+ ) | ( model.labelLike+ ) ),
      ( model.noteLike | model.biblLike )*,
      terrain*
   )
}
```

Example

```
<place  xml:id="KERG">
  <placeName>Kerguelen Islands</placeName>
<!-- ... -->
  <terrain>
    <desc>antarctic tundra</desc>
  </terrain>
<!-- ... -->
</place>
```

<text> contains a single text of any kind, whether unitary or composite, for example a poem or drama, a collection of essays, a novel, a dictionary, or a corpus sample.

Module textstructure — *4. Default Text Structure*

Attributes att.declaring (*@decls*) att.typed (*@type*, *@subtype*)

Used by TEI group

May contain

analysis: interp interpGrp span spanGrp

certainty: certainty respons

core: cb gap index lb milestone note pb

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp timeline

spoken: incident kinesic pause shift vocal writing

textcrit: witDetail

textstructure: back body front group

transcr: addSpan damageSpan delSpan fw space

Declaration

```
element text
{
   att.global.attributes,
   att.declaring.attributes,
   att.typed.attributes,
   (
      model.global*,
      ( front, model.global* )?,
      ( body | group ),
      model.global*,
      ( back, model.global* )?
   )
}
```

Example

```
<text>
  <front>
    <docTitle>
      <titlePart>Autumn Haze</titlePart>
    </docTitle>
  </front>
```

```
  <body>
    <l>Is it a dragonfly or a maple leaf</l>
    <l>That settles softly down upon the water?</l>
  </body>
</text>
```

Example The body of a text may be replaced by a group of nested texts, as in the following schematic:

```
<text>
  <front/>
  <group>
    <text>
      <front/>
      <body/>
      <back/>
    </text>
    <text/>
  </group>
</text>
```

Note This element should not be used to represent a text which is inserted at an arbitrary point within the structure of another, for example as in an embedded or quoted narrative; the <floatingText> is provided for this purpose.

<textClass>

(text classification) groups information which describes the nature or topic of a text in terms of a standard classification scheme, thesaurus, etc.

Module header — *2. The TEI Header*

Attributes att.declarable (*@default*)

Used by model.profileDescPart

May contain

header: catRef classCode keywords

Declaration

```
element textClass
{
   att.global.attributes,
   att.declarable.attributes,
   ( classCode | catRef | keywords )*
}
```

Example

```
<taxonomy>
  <category xml:id="acprose">
    <catDesc>Academic prose</catDesc>
  </category>
<!-- other categories here -->
</taxonomy>
<!-- ... -->
<textClass>
  <catRef target="#acprose"/>
  <classCode scheme="http://www.udcc.org">001.9</classCode>
```

```
  <keywords scheme="http://authorities.loc.gov">
   <list>
    <item>End of the world</item>
    <item>History - philosophy</item>
   </list>
  </keywords>
 </textClass>
```

<textDesc> (text description) provides a description of a text in terms of its situational parameters.

Module corpus — *15. Language Corpora*

Attributes att.declarable (*@default*)

Used by model.catDescPart model.profileDescPart

May contain

corpus: purpose

Declaration

```
element textDesc
{
   att.global.attributes,
   att.declarable.attributes,
   ( model.textDescPart_sequence, purpose+ )
}
```

Example

```
<textDesc n="Informal domestic conversation">
  <channel mode="s"/>
  <constitution type="single"/>
  <derivation type="original"/>
  <domain type="domestic"/>
  <factuality type="mixed"/>
  <interaction type="complete" active="plural" passive="many"/>
  <preparedness type="spontaneous"/>
  <purpose type="entertain" degree="high"/>
  <purpose type="inform" degree="medium"/>
</textDesc>
```

<textLang> (text language) describes the languages and writing systems used by a manuscript (as opposed to its description, which is described in the <langUsage> element).

Module msdescription — *10. Manuscript Description*

Attributes In addition to global attributes

@mainLang (main language) supplies a code which identifies the chief language used in the manuscript.

Status Optional

Datatype `data.language`

Values a recognised language 'tag' generated according to BCP 47 which may additionally be documented by a <language> element in the header

@otherLangs (other languages) one or more codes identifying any other languages used in the manuscript.

Status Optional

Datatype 0–∞ occurrences of `data.language` separated by whitespace

Values a list of codes, each of which is a recognised language 'tag' generated according to BCP 47 which may additionally be documented by a <language> element in the header

Used by msContents msItemStruct model.msItemPart

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address binaryObject cb choice corr date del distinct email emph expan foreign gap gloss graphic hi index lb measure measureGrp mentioned milestone name note num orig pb ptr ref reg rs sic soCalled term time title unclear

dictionaries: lang oRef oVar pRef pVar

figures: formula

gaiji: g

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident specDesc specList tag val

textcrit: app witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element textLang
{
   att.global.attributes,
   attribute mainLang { data.language }?,
   attribute otherLangs { list { data.language* } }?,
   macro.phraseSeq}
```

Example

```
<textLang mainLang="en" otherLangs="la">Predominantly in English with Latin glosses</textLang>
```

<then/> separates the condition from the default in an <if>, or the antecedent and the consequent in a <cond> element.

Module iso-fs — *18. Feature Structures*

Attributes Global attributes only

Used by cond if

May contain Empty element

Declaration `element then { att.global.attributes, empty }`

Example

```
<cond>
  <fs>
    <f name="BAR">
      <symbol value="1"/>
    </f>
  </fs>
  <then/>
  <fs>
    <f name="FOO">
      <binary value="false"/>
    </f>
  </fs>
</cond>
```

Note This element is provided primarily to enhance the human readability of the feature-system declaration.

<time> contains a phrase defining a time of day in any format.

Module core — *3. Elements Available in All TEI Documents*

Attributes att.datableatt.datable.w3c (*@period*, *@when*, *@notBefore*, *@notAfter*, *@from*, *@to*) att.datable.iso (*@when-iso*, *@notBefore-iso*, *@notAfter-iso*, *@from-iso*, *@to-iso*) att.durationatt.duration.w3c (*@dur*) att.duration.iso (*@dur-iso*) att.editLike (*@cert*, *@resp*, *@evidence*, *@source*, *@precision*) att.typed (*@type*, *@subtype*)

Used by model.dateLike

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address binaryObject cb choice corr date del distinct email emph expan foreign gap gloss graphic hi index lb measure measureGrp mentioned milestone name note num orig pb ptr ref reg rs sic soCalled term time title unclear

dictionaries: lang oRef oVar pRef pVar

figures: formula

gaiji: g

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident specDesc specList tag val

textcrit: app witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element time
{
   att.global.attributes,
   att.datable.w3c.attributes,
   att.datable.iso.attributes,
   att.duration.w3c.attributes,
   att.duration.iso.attributes,
   att.editLike.attributes,
   att.typed.attributes,
   ( text | model.gLike | model.phrase | model.global )*
}
```

Example

```
As he sat smiling, the quarter struck —
<time  when="11:45:00">the quarter to twelve</time>.
```

<timeline> (timeline) provides a set of ordered points in time which can be linked to elements of a spoken text to create a temporal alignment of that text.

Module linking — *16. Linking, Segmentation, and Alignment*

Attributes In addition to global attributes

@origin designates the origin of the timeline, i.e. the time at which it begins.

Status **Required**

Datatype `data.pointer`

Values must point either to one of the <when> elements in its content, or to another <timeline> element.

Note If the absolute value for the time of origin is not known, an arbitrary time (such as 00:00) should be used.

@unit specifies the unit of time corresponding to the **interval** value of the timeline or of its constituent points in time.

Status Required when applicable

Datatype `data.enumerated`

Suggested values include: **d** (days)

h (hours)

min (minutes)

s (seconds)

ms (milliseconds)

@interval specifies the numeric portion of a time interval

Status Optional

Datatype `xsd:float { minExclusive = "0" } | "regular" | "irregular"`

Values a positive number, or one of the two special values **irregular** or **regular**.

Note The value **irregular** indicates uncertainty about all the intervals in the timeline; the value **regular** indicates that all the intervals are evenly spaced, but the size of the intervals is not known; numeric values indicate evenly spaced values of the size specified. If individual points in time in the timeline are given different values for the **interval** attribute, those values locally override the value given in the timeline.

Used by model.global.meta

May contain

linking: when

Declaration

```
element timeline
{
   att.global.attributes,
   attribute origin { data.pointer },
   attribute unit { "d" | "h" | "min" | "s" | "ms" | xsd:Name }?,
   attribute interval
   {
      xsd:float { minExclusive = "0" } | "regular" | "irregular"
   }?,
   when+
}
```

Example

```
<timeline xml:id="TL01" origin="#TL-w0" unit="ms">
  <when xml:id="TL-w0" absolute="11:30:00"/>
  <when xml:id="TL-w1" interval="unknown" since="#TL-w0"/>
  <when xml:id="TL-w2" interval="100" since="#TL-w1"/>
  <when xml:id="TL-w3" interval="200" since="#TL-w2"/>
  <when xml:id="TL-w4" interval="150" since="#TL-w3"/>
  <when xml:id="TL-w5" interval="250" since="#TL-w4"/>
  <when xml:id="TL-w6" interval="100" since="#TL-w5"/>
</timeline>
```

Note one or more points in time, one of which is its origin

<title> contains the full title of a work of any kind.

Module core — *3. Elements Available in All TEI Documents*

Attributes In addition to global attributes

@level indicates the bibliographic level for a title, that is, whether it identifies an article, book, journal, series, or unpublished material.

Status Required when applicable

Legal values are: **a** (analytic) analytic title (article, poem, or other item published as part of a larger item)

m (monographic) monographic title (book, collection, or other item published as a distinct item, including single volumes of multi-volume works)

j (journal) journal title

s (series) series title

u (unpublished) title of unpublished material (including theses and dissertations unless published by a commercial press)

Note If the title appears directly enclosed within an <analytic> element, the **level**, if given, must be 'a'; if it appears directly enclosed within a <monogr> element, **level** must be 'm', 'j', or 'u'; when <title> is directly enclosed by <series>, **level** must be 's'. If it appears within a <msItem>, this attribute should not be supplied.

@type classifies the title according to some convenient typology.

Status Optional

Datatype `data.enumerated`

Sample values include: **main** main title

sub (subordinate) subtitle, title of part

alt (alternate) alternate title, often in another language, by which the work is also known

short abbreviated form of title

desc (descriptive) descriptive paraphrase of the work functioning as a title

Note This attribute is provided for convenience in analysing titles and processing them according to their type; where such specialized processing is not necessary, there is no need for such analysis, and the entire title, including subtitles and any parallel titles, may be enclosed within a single <title> element.

Used by analytic monogr msItemStruct series seriesStmt titleStmt model.emphLike model.msItemPart

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address bibl biblStruct binaryObject cb choice cit corr date del desc distinct email emph expan foreign gap gloss graphic hi index label lb list listBibl measure measureGrp mentioned milestone name note num orig pb ptr q quote ref reg rs said sic soCalled stage term time title unclear

dictionaries: lang oRef oVar pRef pVar

drama: camera caption castList move sound tech view

figures: figure formula table

gaiji: g

header: biblFull

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material msDesc origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat

geogName listNym listOrg listPerson listPlace nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att classSpec code eg egXML elementSpec gi ident listRef macroSpec moduleRef moduleSpec specDesc specGrp specGrpRef specList tag val

textcrit: app listWit witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element title
{
   att.global.attributes,
   attribute level { "a" | "m" | "j" | "s" | "u" }?,
   attribute type { data.enumerated }?,
   macro.paraContent}
```

Example

```
<title>La vie mode d'emploi. Romans.</title>
```

Example

```
<title>Synthese: an international journal for epistemology, methodology
and history of science</title>
```

Example

```
<title>Information Technology and the Research Process: Proceedings
of a conference held at Cranfield Institute of Technology, UK, 18–21
July 1989</title>
```

Example

```
<title>Hardy's Tess of the D'Urbervilles: a machine readable edition</title>
```

<titlePage> (title page) contains the title page of a text, appearing within the front or back matter.

Module textstructure — *4. Default Text Structure*

Attributes In addition to global attributes

@type classifies the title page according to any convenient typology.

Status Optional

Datatype `data.enumerated`

Values Any string, e.g. *full*, *half*, *Series*, etc.

Note This attribute allows the same element to be used for volume title pages, series title pages, etc., as well as for the 'main' title page of a work.

Used by model.frontPart

May contain

analysis: interp interpGrp span spanGrp

certainty: certainty respons

core: binaryObject cb gap graphic index lb milestone note pb

figures: figure

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp timeline

spoken: incident kinesic pause shift vocal writing

textcrit: witDetail

textstructure: byline docAuthor docDate docEdition docImprint docTitle epigraph imprimatur titlePart

transcr: addSpan damageSpan delSpan fw space

Declaration

```
element titlePage
{
   att.global.attributes,
   attribute type { data.enumerated }?,
   (
      model.global*,
      ( model.titlepagePart ),
      ( model.titlepagePart | model.global )*
   )
}
```

Example

```
<titlePage>
  <docTitle>
    <titlePart  type="main">THOMAS OF Reading.</titlePart>
    <titlePart  type="alt">OR, The sixe worthy yeomen
        of the West.</titlePart>
  </docTitle>
  <docEdition>Now the fourth time corrected and enlarged</docEdition>
  <byline>By T.D.</byline>
  <figure>
    <head>TP</head>
    <p>Thou shalt labor till thou returne to duste</p>
    <figDesc>Printers Ornament used by TP</figDesc>
  </figure>
  <docImprint>Printed at <name  type="place">London</name>
     for <name>T.P.</name>
    <date>1612.</date>
  </docImprint>
</titlePage>
```

<titlePart> contains a subsection or division of the title of a work, as indicated on a title page.

Module textstructure — *4. Default Text Structure*

Attributes In addition to global attributes

@type specifies the role of this subdivision of the title.

Status Optional

Datatype `data.enumerated`

Suggested values include: **main** main title of the work **[Default]**

sub (subordinate) subtitle of the work

alt (alternate) alternative title of the work

short abbreviated form of title

desc (descriptive) descriptive paraphrase of the work

Used by docTitle model.titlepagePart model.pLike.front

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address bibl biblStruct binaryObject cb choice cit corr date del desc distinct email emph expan foreign gap gloss graphic hi index label lb list listBibl measure measureGrp mentioned milestone name note num orig pb ptr q quote ref reg rs said sic soCalled stage term time title unclear

dictionaries: lang oRef oVar pRef pVar

drama: camera caption castList move sound tech view

figures: figure formula table

gaiji: g

header: biblFull

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material msDesc origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName listNym listOrg listPerson listPlace nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att classSpec code eg egXML elementSpec gi ident listRef macroSpec moduleRef moduleSpec specDesc specGrp specGrpRef specList tag val

textcrit: app listWit witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element titlePart
{
   att.global.attributes,
   attribute type { "main" | "sub" | "alt" | "short" | "desc" | xsd:Name }?,
   macro.paraContent}
```

Example

```
<docTitle>
  <titlePart  type="main">THE FORTUNES
```

```
      AND MISFORTUNES Of the FAMOUS
      Moll Flanders, &c.
   </titlePart>
   <titlePart  type="desc">Who was BORN in NEWGATE,
      And during a Life of continu'd Variety for
      Threescore Years, besides her Childhood, was
      Twelve Year a <hi>Whore</hi>, five times a <hi>Wife</hi> (wherof
      once to her own Brother) Twelve Year a <hi>Thief,</hi>
      Eight Year a Transported <hi>Felon</hi> in <hi>Virginia</hi>,
      at last grew <hi>Rich</hi>, liv'd <hi>Honest</hi>, and died a
   <hi>Penitent</hi>.</titlePart>
</docTitle>
```

<titleStmt> (title statement) groups information about the title of a work and those responsible for its intellectual content.

Module header — *2. The TEI Header*

Attributes Global attributes only

Used by biblFull fileDesc

May contain

core: author editor respStmt title

header: funder principal sponsor

Declaration

```
element titleStmt
{
   att.global.attributes,
   ( title+, ( author | editor | sponsor | funder | principal | respStmt )* )
}
```

Example

```
<titleStmt>
  <title>Capgrave's Life of St. John Norbert: a machine-readable
     transcription</title>
  <respStmt>
    <resp>compiled by</resp>
    <name>P.J. Lucas</name>
  </respStmt>
</titleStmt>
```

<tns> (tense) indicates the grammatical tense associated with a given inflected form in a dictionary.

Module dictionaries — *9. Dictionaries*

Attributes att.lexicographic (*@expand*, *@norm*, *@split*, *@value*, *@orig*, *@location*, *@mergedIn*, *@opt*)

Used by model.morphLike model.entryPart

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address bibl biblStruct binaryObject cb choice cit corr date del desc distinct email emph expan foreign gap gloss graphic hi index label lb list listBibl measure measureGrp mentioned milestone name note num orig pb ptr q quote ref reg rs said sic soCalled stage term time title unclear

dictionaries: lang oRef oVar pRef pVar

drama: camera caption castList move sound tech view

figures: figure formula table

gaiji: g

header: biblFull

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material msDesc origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName listNym listOrg listPerson listPlace nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att classSpec code eg egXML elementSpec gi ident listRef macroSpec moduleRef moduleSpec specDesc specGrp specGrpRef specList tag val

textcrit: app listWit witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element tns
{
   att.global.attributes,
   att.lexicographic.attributes,
   macro.paraContent}
```

Example Taken from *Wörterbuch der Deutschen Sprache. Veranstaltet und herausgegeben von Joachim Heinrich Campe. Vierter Theil. S - bis - T.* (Braunschweig 1810. In der Schulbuchhandlung):

> Treffen, v. unregelm. ... du triffst, ...

```
<entry>
  <form type="inflected">
    <gramGrp>
      <per value="2"/>
      <number value="singular"/>
      <tns value="present"/>
      <mood value="indicative"/>
    </gramGrp>
    <form type="personalpronoun">
      <orth>du</orth>
    </form>
    <form type="headword">
```

```
      <orth>
        <oVar>triffst</oVar>
      </orth>
    </form>
  </form>
</entry>
```

Note This element is synonymous with **<gram type="tense">**.

<trailer> contains a closing title or footer appearing at the end of a division of a text.

Module textstructure — *4. Default Text Structure*

Attributes Global attributes only

Used by castGroup model.divBottomPart

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address binaryObject cb choice corr date del distinct email emph expan foreign gap gloss graphic hi index lb measure measureGrp mentioned milestone name note num orig pb ptr ref reg rs sic soCalled term time title unclear

dictionaries: lang oRef oVar pRef pVar

figures: formula

gaiji: g

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident specDesc specList tag val

textcrit: app witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element trailer { att.global.attributes, macro.phraseSeq }
```

Example

```
<trailer>Explicit pars tertia</trailer>
```

<trait> contains a description of some culturally-determined and in principle unchanging characteristic attributed to a person or place .

Module namesdates — *13. Names, Dates, People, and Places*

Attributes att.datableatt.datable.w3c (*@period*, *@when*, *@notBefore*, *@notAfter*, *@from*, *@to*) att.datable.iso (*@when-iso*, *@notBefore-iso*, *@notAfter-iso*, *@from-iso*, *@to-iso*) att.editLike (*@cert*, *@resp*, *@evidence*, *@source*, *@precision*) att.naming (*@key*, *@ref*, *@nymRef*) att.typed (*@type*, *@subtype*)

Used by trait model.persTraitLike model.placeTraitLike

May contain

core: bibl biblStruct desc head label note p

header: biblFull

linking: ab

msdescription: msDesc

namesdates: trait

textcrit: witDetail

Declaration

```
element trait
{
   att.global.attributes,
   att.datable.w3c.attributes,
   att.datable.iso.attributes,
   att.editLike.attributes,
   att.naming.attributes,
   att.typed.attributes,
   (
      trait+
    | ( model.headLike*, model.pLike+, ( model.noteLike | model.biblLike )* )
    | ( ( model.labelLike | model.noteLike | model.biblLike )* )
   )
}
```

Example

```
<trait
   cert="high"
   type="social"
   from="1987-01-01"
   to="1997-12-31">
  <label>citizenship</label>
  <desc>Between 1987 and 1997 held status of naturalized UK citizen</desc>
</trait>
```

Example

```
<trait  type="physical">
  <label>Eye colour</label>
  <desc>Blue</desc>
</trait>
```

<tree> encodes a tree, which is made up of a root, internal nodes, leaves, and arcs from root to leaves.

Module nets — *19. Graphs, Networks, and Trees*

Attributes In addition to global attributes

@arity gives the maximum number of children of the root and internal nodes of the tree.

Status Optional

Datatype `data.count`

Values A nonnegative integer.

@ord (ordered) indicates whether or not the tree is ordered, or if it is partially ordered.

Status **Required**

Legal values are: **true** indicates that all of the branching nodes of the tree are ordered. **[Default]**

partial indicates that some of the branching nodes of the tree are ordered and some are unordered.

false indicates that all of the branching nodes of the tree are unordered.

@order gives the order of the tree, i.e., the number of its nodes.

Status Optional

Datatype `data.count`

Values A nonnegative integer.

Note The size of a tree is always one less than its order, hence there is no need for both a **size** and **order** attribute.

Used by forest model.divPart

May contain

core: label

nets: iNode leaf root

Declaration

```
element tree
{
   att.global.attributes,
   attribute arity { data.count }?,
   attribute ord { "true" | "partial" | "false" },
   attribute order { data.count }?,
   ( label?, ( ( leaf | iNode )*, root, ( leaf | iNode )* ) )
}
```

Example

```
<tree
   n="ex2"
   arity="2"
   ord="partial"
   order="13">
  <root  xml:id="G-div1"  children="#G-plu1 #G-exp1"  ord="true">
    <label>/</label>
  </root>
  <iNode
     xml:id="G-plu1"
     children="#G-exp2 #G-exp3"
     parent="#G-div1"
     ord="false">
```

```
      <label>+</label>
    </iNode>
    <iNode
       xml:id="G-exp1"
       children="#G-plu2 #G-num2.3"
       parent="#G-div1"
       ord="true">
      <label>**</label>
    </iNode>
    <iNode
       xml:id="G-exp2"
       children="#G-vara1 #G-num2.1"
       parent="#G-plu1"
       ord="true">
      <label>**</label>
    </iNode>
    <iNode
       xml:id="G-exp3"
       children="#G-varb1 #G-num2.2"
       parent="#G-plu1"
       ord="true">
      <label>**</label>
    </iNode>
    <iNode
       xml:id="G-plu2"
       children="#G-vara2 #G-varb2"
       parent="#G-exp1"
       ord="false">
      <label>+</label>
    </iNode>
    <leaf  xml:id="G-vara1"  parent="#G-exp2">
      <label>a</label>
    </leaf>
    <leaf  xml:id="G-num2.1"  parent="#G-exp2">
      <label>2</label>
    </leaf>
    <leaf  xml:id="G-varb1"  parent="#G-exp3">
      <label>b</label>
    </leaf>
    <leaf  xml:id="G-num2.2"  parent="#G-exp3">
      <label>2</label>
    </leaf>
    <leaf  xml:id="G-vara2"  parent="#G-plu2">
      <label>a</label>
    </leaf>
    <leaf  xml:id="G-varb2"  parent="#G-plu2">
      <label>b</label>
    </leaf>
    <leaf  xml:id="G-num2.3"  parent="#G-exp1">
      <label>2</label>
    </leaf>
  </tree>
```

Note A root, and zero or more internal nodes and leaves, but if there is an internal node, there must also be at least one leaf.

<triangle> (underspecified embedding tree, so called because of its characteristic shape when drawn)
Provides for an underspecified eTree, that is, an eTree with information left out.

Module nets — *19. Graphs, Networks, and Trees*

Attributes In addition to global attributes

@value provides the value of a triangle, which is the identifier of a feature structure or other analytic element.

Status Required when applicable

Datatype `data.pointer`

Values A valid identifier of a feature structure or other analytic element.

Used by eTree forest triangle

May contain

core: label

nets: eLeaf eTree triangle

Declaration

```
element triangle
{
   att.global.attributes,
   attribute value { data.pointer }?,
   ( label?, ( eTree | triangle | eLeaf )* )
}
```

Example

```
<triangle>
  <label>NP</label>
  <eLeaf>
    <label>the periscope</label>
  </eLeaf>
</triangle>
```

Note An optional label followed by zero or more embedding trees, triangles, or embedding leafs.

<u> (utterance) a stretch of speech usually preceded and followed by silence or by a change of speaker.

Module spoken — *8. Transcriptions of Speech*

Attributes att.timed (*@start*, *@end*) att.duration.w3c (*@dur*) att.declaring (*@decls*) att.ascribed (*@who*)

@trans (transition) indicates the nature of the transition between this utterance and the previous one.

Status Optional

Legal values are: **smooth** this utterance begins without unusual pause or rapidity. **[Default]**

latching this utterance begins with a markedly shorter pause than normal.

overlap this utterance begins before the previous one has finished.

pause this utterance begins after a noticeable pause.

Used by model.divPart.spoken

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address binaryObject cb choice corr date del distinct email emph expan foreign gap gloss graphic hi index lb measure measureGrp mentioned milestone name note num orig pb ptr ref reg rs sic soCalled term time title unclear

dictionaries: lang oRef oVar pRef pVar

figures: formula

gaiji: g

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident specDesc specList tag val

textcrit: app witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element u
{
   att.global.attributes,
   att.timed.attributes,
   att.duration.w3c.attributes,
   att.declaring.attributes,
   att.ascribed.attributes,
   attribute trans { "smooth" | "latching" | "overlap" | "pause" }?,
   ( text | model.gLike | model.phrase | model.global )*
}
```

Example

```
<u  who="#spkr1">if did you set</u>
<u  trans="latching"  who="#spkr2">well Joe and I set it between us</u>
<list  type="speakers">
  <item  xml:id="spkr1"/>
  <item  xml:id="spkr2"/>
</list>
```

Note Prose and a mixture of speech elementsAlthough individual transcriptions may consistently use <u> elements for turns or other units, and although in most cases a <u> will be delimited by pause or change of speaker, <u> is not required to represent a turn or any communicative event, nor to be bounded by pauses or change of speaker. At a minimum, a <u> is some phonetic production by a given speaker.

<unclear> contains a word, phrase, or passage which cannot be transcribed with certainty because it is illegible or inaudible in the source.

Module core — *3. Elements Available in All TEI Documents*

Attributes att.editLike (*@cert*, *@resp*, *@evidence*, *@source*, *@precision*)

@reason indicates why the material is hard to transcribe.

Status Optional

Datatype 1–∞ occurrences of `data.word` separated by whitespace

Values one or more words describing the difficulty, e.g. *faded*, *background noise*, *passing truck*, *illegible*, *eccentric ductus*.

```
<div>
  <head>Rx</head>
  <p>500 mg <unclear
       reason="illegible">placebo</unclear>
  </p>
</div>
```

@hand Where the difficulty in transcription arises from action (partial deletion, etc.) assignable to an identifiable hand, signifies the hand responsible for the action.

Status Optional

Datatype `data.pointer`

Values must be one of the hand identifiers declared in the document header (see section *11.4.1. Document Hands*).

@agent Where the difficulty in transcription arises from damage, categorizes the cause of the damage, if it can be identified.

Status Optional

Datatype `data.enumerated`

Sample values include: **rubbing** damage results from rubbing of the leaf edges

mildew damage results from mildew on the leaf surface

smoke damage results from smoke

Used by model.pPart.transcriptional model.choicePart

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address bibl biblStruct binaryObject cb choice cit corr date del desc distinct email emph expan foreign gap gloss graphic hi index label lb list listBibl measure measureGrp mentioned milestone name note num orig pb ptr q quote ref reg rs said sic soCalled stage term time title unclear

dictionaries: lang oRef oVar pRef pVar

drama: camera caption castList move sound tech view

figures: figure formula table

gaiji: g

header: biblFull

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material msDesc origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName listNym listOrg listPerson listPlace nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att classSpec code eg egXML elementSpec gi ident listRef macroSpec moduleRef moduleSpec specDesc specGrp specGrpRef specList tag val

textcrit: app listWit witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element unclear
{
   att.global.attributes,
   att.editLike.attributes,
   attribute reason { list { data.word+ } }?,
   attribute hand { data.pointer }?,
   attribute agent { data.enumerated }?,
   macro.paraContent}
```

Note The same element is used for all cases of uncertainty in the transcription of element content, whether for written or spoken material. For other aspects of certainty, uncertainty, and reliability of tagging and transcription, see chapter *21. Certainty and Responsibility*.The <damage>, <gap>, <del>, <unclear> and <supplied> elements may be closely allied in use. See section *11.5.2. Use of the <gap>, <del>, <damage>, <unclear>, and <supplied> Elements in Combination* for discussion of which element is appropriate for which circumstance.

<unicodeName>

<unicodeName> (unicode property name) contains the name of a registered Unicode normative or informative property.

Module gaiji — *5. Representation of Non-standard Characters and Glyphs*

Attributes In addition to global attributes

@version specifies the version number of the Unicode Standard in which this property name is defined.

Status Optional

Datatype `data.numeric`

Values a valid version number.

Used by charProp

May contain Character data only

Declaration

```
element unicodeName
{
   att.global.attributes,
   attribute version { data.numeric }?,
```

```
    text
}
```

Example

```
<unicodeName>character-decomposition-mapping</unicodeName>
<unicodeName>general-category</unicodeName>
```

Note A definitive list of current Unicode property names is provided in The Unicode Standard.

<usg> (usage) contains usage information in a dictionary entry.

Module dictionaries — *9. Dictionaries*

Attributes att.lexicographic (*@expand*, *@norm*, *@split*, *@value*, *@orig*, *@location*, *@mergedIn*, *@opt*)

@type classifies the usage information using any convenient typology.

Status Optional

Datatype `data.enumerated`

Sample values include: **geo** (geographic) geographic area

time temporal, historical era (archaic, old, etc.)

dom (domain) domain or subject matter (e.g. scientific, literary etc.)

reg (register)

style style (figurative, literal, etc.)

plev (preference level) preference level (chiefly, usually, etc.)

lang (language) name of a language mentioned in etymological or other linguistic discussion.

gram (grammatical) grammatical usage

syn (synonym) synonym given to show use

hyper (hypernym) hypernym given to show usage

colloc (collocation) contains a collocate of the headword.

comp (complement) typical complement

obj (object) typical object

subj (subject) typical subject

verb typical verb

hint unclassifiable piece of information to guide sense choice

Used by etym xr model.entryPart.top model.entryPart model.gramPart

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address bibl biblStruct binaryObject cb choice cit corr date del desc distinct email emph expan foreign gap gloss graphic hi index label lb list listBibl measure measureGrp mentioned milestone name note num orig pb ptr q quote ref reg rs said sic soCalled stage term time title unclear

dictionaries: lang oRef oVar pRef pVar

drama: camera caption castList move sound tech view

figures: figure formula table

gaiji: g

header: biblFull

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material msDesc origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName listNym listOrg listPerson listPlace nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att classSpec code eg egXML elementSpec gi ident listRef macroSpec moduleRef moduleSpec specDesc specGrp specGrpRef specList tag val

textcrit: app listWit witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element usg
{
   att.global.attributes,
   att.lexicographic.attributes,
   attribute type { data.enumerated }?,
   macro.paraContent}
```

Example

```
<form>
  <orth>colour</orth>
  <form>
    <usg  type="geo">U.S.</usg>
    <orth>color</orth>
  </form>
</form>
```

<vAlt> (value alternation) represents the value part of a feature-value specification which contains a set of values, only one of which can be valid.

Module iso-fs — *18. Feature Structures*

Attributes Global attributes only

Used by model.featureVal.single

May contain

iso-fs: binary default fs numeric string symbol vAlt vColl vLabel vMerge vNot

Declaration

```
element vAlt
{
```

```
   att.global.attributes,
   ( ( model.featureVal ), model.featureVal+ )
}
```

Example

```
<f  name="gender">
  <vAlt>
    <symbol  value="masculine"/>
    <symbol  value="neuter"/>
    <symbol  value="feminine"/>
  </vAlt>
</f>
```

<vColl> (collection of values) represents the value part of a feature-value specification which contains multiple values organized as a set, bag, or list.

Module iso-fs — *18. Feature Structures*

Attributes In addition to global attributes

@org (organization) indicates organization of given value or values as set, bag or list.

Status Required when applicable

Legal values are: **set** indicates that the given values are organized as a set.

bag indicates that the given values are organized as a bag (multiset).

list indicates that the given values are organized as a list.

Used by model.featureVal.complex

May contain

iso-fs: binary default fs numeric string symbol vAlt vLabel

Declaration

```
element vColl
{
   att.global.attributes,
   attribute org { "set" | "bag" | "list" }?,
   ( ( fs | model.featureVal.single )* )
}
```

Example

```
<f  name="name">
  <vColl>
    <string>Jean</string>
    <string>Luc</string>
    <string>Godard</string>
  </vColl>
</f>
```

Example

```
<fs>
  <f  name="lex">
    <symbol  value="auxquels"/>
```

```
  </f>
  <f  name="maf">
    <vColl  org="list">
      <fs>
        <f  name="cat">
          <symbol  value="prep"/>
        </f>
      </fs>
      <fs>
        <f  name="cat">
          <symbol  value="pronoun"/>
        </f>
        <f  name="kind">
          <symbol  value="rel"/>
        </f>
        <f  name="num">
          <symbol  value="pl"/>
        </f>
        <f  name="gender">
          <symbol  value="masc"/>
        </f>
      </fs>
    </vColl>
  </f>
</fs>
```

<vDefault> (value default) declares the default value to be supplied when a feature structure does not contain an instance of <f> for this name; if unconditional, it is specified as one (or, depending on the value of the **org** attribute of the enclosing <fDecl>) more <fs> elements or primitive values; if conditional, it is specified as one or more <if> elements; if no default is specified, or no condition matches, the value **none** is assumed.

Module iso-fs — *18. Feature Structures*

Attributes Global attributes only

Used by fDecl

May contain

iso-fs: binary default fs if numeric string symbol vAlt vColl vLabel vMerge vNot

Declaration

```
element vDefault { att.global.attributes, ( model.featureVal+ | if+ ) }
```

Example

```
<fDecl  name="INV">
  <fDescr>inverted sentence</fDescr>
  <vRange>
    <vAlt>
      <binary  value="true"/>
      <binary  value="false"/>
    </vAlt>
  </vRange>
  <vDefault>
```

```
      <binary value="false"/>
    </vDefault>
  </fDecl>
```

Note May contain a legal feature value, or a series of <if> elements.

<vLabel> (value label) represents the value part of a feature-value specification which appears at more than one point in a feature structure.

Module iso-fs — *18. Feature Structures*

Attributes In addition to global attributes

@name supplies a name for the sharing point.

Status **Required**

Datatype `data.word`

Values An identifying name.

Used by model.featureVal.single

May contain

iso-fs: binary default fs numeric string symbol vAlt vColl vLabel vMerge vNot

Declaration

```
element vLabel
{
   att.global.attributes,
   attribute name { data.word },
   model.featureVal?
}
```

Example

```
<fs>
  <f name="nominal">
    <fs>
      <f name="nm-num">
        <vLabel name="L1">
          <symbol value="singular"/>
        </vLabel>
      </f>
<!-- other nominal features -->
    </fs>
  </f>
  <f name="verbal">
    <fs>
      <f name="vb-num">
        <vLabel name="L1"/>
      </f>
    </fs>
<!-- other verbal features -->
  </f>
</fs>
```

<vMerge> (merged collection of values) represents a feature value which is the result of merging together the feature values contained by its children, using the organization specified by the **org** attribute.

Module iso-fs — *18. Feature Structures*

Attributes In addition to global attributes

@org indicates the organization of the resulting merged values as set, bag or list.

Status Required when applicable

Legal values are: **set** indicates that the resulting values are organized as a set.

bag indicates that the resulting values are organized as a bag (multiset).

list indicates that the resulting values are organized as a list.

Used by model.featureVal.complex

May contain

iso-fs: binary default fs numeric string symbol vAlt vColl vLabel vMerge vNot

Declaration

```
element vMerge
{
   att.global.attributes,
   attribute org { "set" | "bag" | "list" }?,
   model.featureVal+
}
```

Example

```
<vMerge org="list">
  <vColl org="set">
    <symbol value="masculine"/>
    <symbol value="neuter"/>
    <symbol value="feminine"/>
  </vColl>
  <symbol value="indeterminate"/>
</vMerge>
```

This example returns a list, concatenating the indeterminate value with the set of values masculine, neuter and feminine.

<vNot> (value negation) represents a feature value which is the negation of its content.

Module iso-fs — *18. Feature Structures*

Attributes Global attributes only

Used by model.featureVal.complex

May contain

iso-fs: binary default fs numeric string symbol vAlt vColl vLabel vMerge vNot

Declaration

```
element vNot { att.global.attributes, ( model.featureVal ) }
```

Example

```
<vNot>
  <symbol value="masculine"/>
</vNot>
```

Example

```
<f name="mode">
  <vNot>
    <vAlt>
      <symbol value="infinitive"/>
      <symbol value="participle"/>
    </vAlt>
  </vNot>
</f>
```

<vRange> (value range) defines the range of allowed values for a feature, in the form of an <fs>, <vAlt>, or primitive value; for the value of an <f> to be valid, it must be subsumed by the specified range; if the <f> contains multiple values (as sanctioned by the **org** attribute), then each value must be subsumed by the <vRange>.

Module iso-fs — *18. Feature Structures*

Attributes Global attributes only

Used by fDecl

May contain

iso-fs: binary default fs numeric string symbol vAlt vColl vLabel vMerge vNot

Declaration

```
element vRange { att.global.attributes, model.featureVal }
```

Example

```
<fDecl name="INV">
  <fDescr>inverted sentence</fDescr>
  <vRange>
    <vAlt>
      <binary value="true"/>
      <binary value="false"/>
    </vAlt>
  </vRange>
  <vDefault>
    <binary value="false"/>
  </vDefault>
</fDecl>
```

Note May contain any legal feature-value specification.

<val> (value) contains a single attribute value.

Module tagdocs — *22. Documentation Elements*

Attributes Global attributes only

Used by model.phrase.xml

May contain Character data only

Declaration
```
element val { att.global.attributes, text }
```

Example
```
<val>unknown</val>
```

<valDesc> (value description) specifies any semantic or syntactic constraint on the value that an attribute may take, additional to the information carried by the datatype element.

Module tagdocs — *22. Documentation Elements*

Attributes In addition to global attributes

@mode specifies the effect of this declaration on its parent module.

Status Optional

Legal values are: **add** this declaration is added to the current definitions **[Default]**

delete this declaration and all of its children are removed from the current setup

change this declaration changes the declaration of the same name in the current definition

replace this declaration replaces the declaration of the same name in the current definition

Used by attDef

May contain

analysis: interp interpGrp span spanGrp

certainty: certainty respons

core: abbr address cb choice date distinct email emph expan foreign gap gloss index lb measure measureGrp mentioned milestone name note num pb ptr ref rs soCalled term time title

dictionaries: lang

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident tag val

textcrit: witDetail

transcr: addSpan am damageSpan delSpan ex fw handShift space subst

Declaration

```
element valDesc
{
   att.global.attributes,
```

```
   attribute mode { "add" | "delete" | "change" | "replace" }?,
   macro.phraseSeq.limited}
```

Example

```
<valDesc>must point to another <gi>align</gi>
element logically preceding this one.</valDesc>
```

<valItem> documents a single attribute-value within a list of possible or mandatory items.

Module tagdocs — *22. Documentation Elements*

Attributes att.identified (*@ident*, *@predeclare*, *@module*, *@mode*)

Used by valList

May contain

core: desc gloss

tagdocs: altIdent equiv

Declaration

```
element valItem
{
   att.global.attributes,
   att.identified.attributes,
   model.glossLike*
}
```

Example

```
<valItem  ident="dub">
  <altIdent  xml:lang="fr">dou</altIdent>
  <equiv  name="unknown"/>
  <gloss>dubious</gloss>
  <desc>used when the application of this element is doubtful or uncertain</desc>
</valItem>
```

<valList> (value list) contains one or more <valItem> elements defining possible values for an attribute.

Module tagdocs — *22. Documentation Elements*

Attributes In addition to global attributes

@mode specifies the effect of this declaration on its parent module.

Status Optional

Legal values are: **add** this declaration is added to the current definitions [**Default**]

delete this declaration and all of its children are removed from the current setup

change this declaration changes the declaration of the same name in the current definition

replace this declaration replaces the declaration of the same name in the current definition

@type specifies the extensibility of the list of attribute values specified.

Status Optional

Legal values are: **closed** only the values specified are permitted.

semi (semi-open) all the values specified should be supported, but other values are legal and software should have appropriate fallback processing for them.

open the values specified are sample values only. [**Default**]

Used by attDef content

May contain

tagdocs: valItem

Declaration

```
element valList
{
   att.global.attributes,
   attribute mode { "add" | "delete" | "change" | "replace" }?,
   attribute type { "closed" | "semi" | "open" }?,
   valItem+
}
```

Example

```
<valList type="closed">
  <valItem ident="req">
    <gloss>required</gloss>
  </valItem>
  <valItem ident="mwa">
    <gloss>mandatory when applicable</gloss>
  </valItem>
  <valItem ident="rec">
    <gloss>recommended</gloss>
  </valItem>
  <valItem ident="rwa">
    <gloss>recommended when applicable</gloss>
  </valItem>
  <valItem ident="opt">
    <gloss>optional</gloss>
  </valItem>
</valList>
```

<value>

(value) contains a single value for some property, attribute, or other analysis.

Module gaiji — *5. Representation of Non-standard Characters and Glyphs*

Attributes Global attributes only

Used by charProp

May contain

gaiji: g

Declaration

```
element value { att.global.attributes, macro.xtext }
```

Example

```
<value>unknown</value>
```

<variantEncoding/> declares the method used to encode text-critical variants.

Module textcrit — *12. Critical Apparatus*

Attributes In addition to global attributes

@method indicates which method is used to encode the apparatus of variants.

Status **Required**

Legal values are: **location-referenced** apparatus uses line numbers or other canonical reference scheme referenced in a base text.

double-end-point apparatus indicates the precise locations of the beginning and ending of each lemma relative to a base text.

parallel-segmentation alternate readings of a passage are given in parallel in the text; no notion of a base text is necessary.

Note The value 'parallel-segmentation' requires in-line encoding of the apparatus.

@location indicates whether the apparatus appears within the running text or external to it.

Status **Required**

Legal values are: **internal** apparatus appears within the running text.

external apparatus appears outside the base text.

Note The value 'external' is inconsistent with the parallel-segmentation method of apparatus markup.

Used by model.encodingPart

May contain Empty element

Declaration

```
element variantEncoding
{
   att.global.attributes,
   attribute method
   {
      "location-referenced" | "double-end-point" | "parallel-segmentation"
   },
   attribute location { "internal" | "external" },
   empty
}
```

Example

```
<variantEncoding method="location-referenced" location="external"/>
```

<view> describes the visual context of some part of a screen play in terms of what the spectator sees, generally independent of any dialogue.

Module drama — *7. Performance Texts*

Attributes Global attributes only

Used by model.stageLike

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address bibl biblStruct binaryObject cb choice cit corr date del desc distinct email emph expan foreign gap gloss graphic hi index l label lb lg list listBibl measure measureGrp mentioned milestone name note num orig p pb ptr q quote ref reg rs said sic soCalled sp stage term time title unclear

dictionaries: lang oRef oVar pRef pVar

drama: camera caption castList move sound tech view

figures: figure formula table

gaiji: g

header: biblFull

iso-fs: fLib fs fvLib

linking: ab alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material msDesc origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName listNym listOrg listPerson listPlace nameLink offset orgName persName placeName region roleName settlement state surname

nets: eTree forest forestGrp graph tree

spoken: incident kinesic pause shift u vocal writing

tagdocs: att classSpec code eg egXML elementSpec gi ident listRef macroSpec moduleRef moduleSpec schemaSpec specDesc specGrp specGrpRef specList tag val

textcrit: app listWit witDetail

textstructure: floatingText

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element view { att.global.attributes, macro.specialPara }
```

Example

```
<view>
  <name>Max</name> joins his daughter
at the window. <hi>Rain</hi> sprays his
face--
</view>
<view>
  <camera>Max's POV</camera> He sees occasional
windows open, and just across from his apartment
house, a <hi>man</hi> opens the front door of
a brownstone--
</view>
```

Example

```
<div type="shot">
  <view>BBC World symbol</view>
  <sp>
    <speaker>Voice Over</speaker>
    <p>Monty Python's Flying Circus tonight comes to you live
        from the Grillomat Snack Bar, Paignton.</p>
  </sp>
</div>
<div type="shot">
  <view>Interior of a nasty snack bar. Customers around, preferably
     real people. Linkman sitting at one of the plastic tables.</view>
  <sp>
    <speaker>Linkman</speaker>
    <p>Hello to you live from the Grillomat Snack Bar.
    </p>
  </sp>
</div>
```

Note A view is a particular form of stage direction.

<vocal> any vocalized but not necessarily lexical phenomenon, for example voiced pauses, non-lexical backchannels, etc.

Module spoken — *8. Transcriptions of Speech*

Attributes att.timed (*@start*, *@end*) att.duration.w3c (*@dur*) att.ascribed (*@who*) att.typed (*@type*, *@subtype*)

@iterated indicates whether or not the phenomenon is repeated.

Status Optional

Datatype `data.xTruthValue`

Note The value **true** indicates that the vocal effect is repeated several times rather than just occurring once.

Used by model.global.spoken

May contain

core: desc gloss

tagdocs: altIdent equiv

Declaration

```
element vocal
{
   att.global.attributes,
   att.timed.attributes,
   att.duration.w3c.attributes,
   att.ascribed.attributes,
   att.typed.attributes,
   attribute iterated { data.xTruthValue }?,
   model.glossLike*
}
```

Example

```
<vocal dur="PT12S">
  <desc>whistles</desc>
</vocal>
<vocal iterated="true">
  <desc>whistles intermittently</desc>
</vocal>
```

<w> (word) represents a grammatical (not necessarily orthographic) word.

Module analysis — *17. Simple Analytic Mechanisms*

Attributes att.segLike (*@type*, *@function*, *@part*) att.metrical (*@met*, *@real*, *@rhyme*)

@lemma provides a lemma for the word, such as an uninflected dictionary entry form.

Status Optional

Datatype `data.key`

@lemmaRef provides a pointer to a definition for the root form of this word form.

Status Optional

Datatype `data.pointer`

Values any valid URI

Used by model.segLike

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: cb gap index lb milestone note pb

gaiji: g

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

spoken: incident kinesic pause shift vocal writing

textcrit: witDetail

transcr: addSpan damageSpan delSpan fw space

Declaration

```
element w
{
   att.global.attributes,
   att.segLike.attributes,
   att.metrical.attributes,
   attribute lemma { data.key }?,
   attribute lemmaRef { data.pointer }?,
   ( text | model.gLike | model.segLike | model.global )*
}
```

Example

```
<w
   type="verb"
   lemma="hit"
```

```
      lemmaRef="http://www.example.com/lexicon/hitvb.xml">hitt<m  type="suffix">ing</m>
    </w>
```

<watermark> contains a word or phrase describing a watermark or similar device.

Module msdescription — *10. Manuscript Description*

Attributes Global attributes only

Used by model.pPart.msdesc

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address binaryObject cb choice corr date del distinct email emph expan foreign gap gloss graphic hi index lb measure measureGrp mentioned milestone name note num orig pb ptr ref reg rs sic soCalled term time title unclear

dictionaries: lang oRef oVar pRef pVar

figures: formula

gaiji: g

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident specDesc specList tag val

textcrit: app witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element watermark { att.global.attributes, macro.phraseSeq }
```

Example

```
<support>
  <p>
    <material>Rag paper</material> with <watermark>anchor</watermark> watermark</p>
</support>
```

<when/> indicates a point in time either relative to other elements in the same timeline tag, or absolutely.

Module linking — *16. Linking, Segmentation, and Alignment*

Attributes In addition to global attributes

@absolute supplies an absolute value for the time.

Status Required when applicable

Datatype `data.temporal.w3c`

Values Times may be given in standard form, as specified in the Encoding Declarations section of the header.

Note Required for the element designated as the value of the **origin** attribute in the **<timeline>** tag.

@unit specifies the unit of time in which the **interval** value is expressed, if this is not inherited from the parent <timeline>.

Status Required when applicable

Datatype `data.enumerated`

Suggested values include: **d** (days)

h (hours)

min (minutes)

s (seconds)

ms (milliseconds)

@interval specifies the numeric portion of a time interval

Status Required when applicable

Datatype `xsd:float { minExclusive = "0" } | "unknown"`

Values a positive number, or the special value **unknown**.

Note The value **unknown** indicates uncertainty about the interval.

@since identifies the reference point for determining the time of the current <when> element, which is obtained by adding the interval to the time of the reference point.

Status Required when applicable

Datatype `data.pointer`

Values Should point to another <when> element in the same <timeline>.

Note If this attribute is omitted, and the **absolute** attribute is not specified, then the reference point is understood to be the origin of the enclosing <timeline> tag.

Used by timeline

May contain Empty element

Declaration

```
element when
{
   att.global.attributes,
   attribute absolute { data.temporal.w3c }?,
   attribute unit { "d" | "h" | "min" | "s" | "ms" | xsd:Name }?,
   attribute interval { xsd:float { minExclusive = "0" } | "unknown" }?,
   attribute since { data.pointer }?,
   empty
}
```

Example

```
<when xml:id="TW3" interval="20" since="#w2"/>
```

Note On this element, the global **xml:id** attribute must be supplied to specify an identifier for this point in time. The value used may be chosen freely provided that it is unique within the document and is a syntactically valid name. There is no requirement for values containing numbers to be in sequence.

<width> contains a measurement measured along the axis perpendicular to the spine.

Module msdescription — *10. Manuscript Description*

Attributes att.dimensions (*@unit*, *@quantity*, *@precision*, *@scope*)

Used by dimensions model.measureLike

May contain

gaiji: g

Declaration

```
element width { att.global.attributes, att.dimensions.attributes, macro.xtext }
```

Example

```
<width unit="in">4</width>
```

<wit> contains a list of one or more sigla of witnesses attesting a given reading, in a textual variation.

Module textcrit — *12. Critical Apparatus*

Attributes att.rdgPart (*@wit*)

Used by app rdgGrp model.rdgPart

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address binaryObject cb choice corr date del distinct email emph expan foreign gap gloss graphic hi index lb measure measureGrp mentioned milestone name note num orig pb ptr ref reg rs sic soCalled term time title unclear

dictionaries: lang oRef oVar pRef pVar

figures: formula

gaiji: g

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident specDesc specList tag val

textcrit: app witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element wit { att.global.attributes, att.rdgPart.attributes, macro.phraseSeq }
```

Example

```
<rdg  wit="#El #Hg">Experience</rdg>
<wit>Ellesmere, Hengwryt</wit>
```

Note This element represents the same information as that provided by the **wit** attribute of the reading; it may be used to record the exact form of the sigla given in the source edition, when that is of interest.

<witDetail> (witness detail) gives further information about a particular witness, or witnesses, to a particular reading.

Module textcrit — *12. Critical Apparatus*

Attributes att.placement (*@place*)

@target indicates the identifier for the reading, or readings, to which the witness detail refers.

Status **Required**

Datatype 1–∞ occurrences of `data.pointer` separated by whitespace

Values the identifier of the reading or readings.

@resp (responsible party) identifies the individual responsible for identifying the witness

Status Optional

Datatype `data.pointer`

Values a pointer to one of the identifiers declared in the document header, associated with a person asserted as responsible for some aspect of the text's creation, transcription, editing, or encoding

@wit (witnesses) indicates the sigil or sigla for the witnesses to which the detail refers.

Status **Required**

Datatype `data.pointer`

Values the identifier of the sigil or sigla.

@type describes the type of information given about the witness.

Status Optional

Datatype `data.enumerated`

Values Values can be taken from any convenient typology of annotation suitable to the work in hand; e.g. **letter_form**, **ornament**, …

Used by model.noteLike

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address binaryObject cb choice corr date del distinct email emph expan foreign gap gloss graphic hi index lb measure measureGrp mentioned milestone name note num orig pb ptr ref reg rs sic soCalled term time title unclear

dictionaries: lang oRef oVar pRef pVar

figures: formula

gaiji: g

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att code gi ident specDesc specList tag val

textcrit: app witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element witDetail
{
   att.global.attributes,
   att.placement.attributes,
   attribute target { list { data.pointer+ } },
   attribute resp { data.pointer }?,
   attribute wit { data.pointer },
   attribute type { data.enumerated }?,
   macro.phraseSeq}
```

Example

```
<app type="substantive">
  <rdgGrp type="subvariants">
    <lem xml:id="W026x" wit="#El #HG">Experience</lem>
    <rdg wit="#Ha4">Experiens</rdg>
  </rdgGrp>
</app>
<witDetail target="#W026x" resp="#PR" wit="#El">Ornamental capital.</witDetail>
```

Note The <witDetail> element should be regarded as a specialized type of <note> element; it is synonymous with **<note type='witnessDetail'>**. It differs from the general purpose <note> in the omission of some attributes seldom applicable to notes within critical apparatus, and in the provision of the **wit** attribute, which permits an application to extract all annotation concerning a particular witness or witnesses from the apparatus.

<witEnd/> (fragmented witness end) indicates the end, or suspension, of the text of a fragmentary witness.

Module textcrit — *12. Critical Apparatus*

Attributes att.rdgPart (*@wit*)

Used by model.rdgPart

May contain Empty element

Declaration

```
element witEnd { att.global.attributes, att.rdgPart.attributes, empty }
```

Example

```
<app>
  <lem  wit="#El #Hg">Experience</lem>
  <rdg  wit="#Ha4">Ex<g  ref="#per"/>
    <witEnd/>
  </rdg>
</app>
```

<witStart/> (fragmented witness start) indicates the beginning, or resumption, of the text of a fragmentary witness.

Module textcrit — *12. Critical Apparatus*

Attributes att.rdgPart (*@wit*)

Used by model.rdgPart

May contain Empty element

Declaration

```
element witStart { att.global.attributes, att.rdgPart.attributes, empty }
```

Example

```
<app>
  <lem  wit="#El #Hg">Auctoritee</lem>
  <rdg  wit="#La #Ra2">auctorite</rdg>
  <rdg  wit="#X">
    <witStart/>auctorite</rdg>
</app>
```

<witness> contains either a description of a single witness referred to within the critical apparatus, or a list of witnesses which is to be referred to by a single sigil.

Module textcrit — *12. Critical Apparatus*

Attributes Global attributes only

Used by listWit

May contain

core: abbr address bibl biblStruct choice cit date desc distinct email emph expan foreign gloss label list listBibl measure measureGrp mentioned name num ptr q quote ref rs said soCalled stage term time title

dictionaries: lang

drama: camera caption castList move sound tech view

figures: figure table

header: biblFull

msdescription: catchwords depth dimensions height heraldry locus material msDesc origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName listNym listOrg listPerson listPlace nameLink offset orgName persName placeName region roleName settlement state surname

tagdocs: att classSpec code eg egXML elementSpec gi ident listRef macroSpec moduleRef moduleSpec specGrp specGrpRef tag val

textcrit: listWit

transcr: am ex handShift subst

Declaration

```
element witness { att.global.attributes, macro.limitedContent }
```

Example

```
<listWit>
  <witness xml:id="EL">Ellesmere, Huntingdon Library 26.C.9</witness>
  <witness xml:id="HG">Hengwrt, National Library of Wales,
    Aberystwyth, Peniarth 392D</witness>
  <witness xml:id="RA2">Bodleian Library Rawlinson Poetic 149
    (see further <ptr target="#MSRP149"/>)</witness>
</listWit>
```

Note The content of the <witness> element may give bibliographic information about the witness or witness group, or it may be empty.

<writing> a passage of written text revealed to participants in the course of a spoken text.

Module spoken — *8. Transcriptions of Speech*

Attributes att.ascribed (*@who*) att.typed (*@type*, *@subtype*) att.timed (*@start*, *@end*) att.duration.w3c (*@dur*)

@source points to a bibliographic citation in the header giving a full description of the source or script of the writing.

Status Optional

Datatype `data.code`

Values Must be a valid identifier for a bibliographic element in the TEI header

@gradual indicates whether the writing is revealed all at once or gradually.

Status Optional

Datatype `data.xTruthValue`

Note The value **true** indicates the writing is revealed gradually; the value **false** that the writing is revealed all at once.

Used by model.global.spoken

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address bibl biblStruct binaryObject cb choice cit corr date del desc distinct email emph expan foreign gap gloss graphic hi index label lb list listBibl measure measureGrp mentioned milestone name note num orig pb ptr q quote ref reg rs said sic soCalled stage term time title unclear

dictionaries: lang oRef oVar pRef pVar

drama: camera caption castList move sound tech view

figures: figure formula table

gaiji: g

header: biblFull

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material msDesc origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName listNym listOrg listPerson listPlace nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att classSpec code eg egXML elementSpec gi ident listRef macroSpec moduleRef moduleSpec specDesc specGrp specGrpRef specList tag val

textcrit: app listWit witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element writing
{
   att.global.attributes,
   att.ascribed.attributes,
   att.typed.attributes,
   att.timed.attributes,
   att.duration.w3c.attributes,
   attribute source { data.code }?,
   attribute gradual { data.xTruthValue }?,
   macro.paraContent}
```

Example

```
<!-- ... --><l>man in a coonskin cap</l>
<writing>coonskin</writing>
<l>in a pig pen</l>
<writing>pig pen</writing>
<l>wants eleven dollar bills</l>
<writing>20 dollar bills</writing>
<l>you only got ten</l>
```

```
<writing>10</writing>
<!-- ... -->
```

Note The <writing> element will usually be short and most simply transcribed as a character string; the content model also allows a sequence of paragraphs and paragraph-level elements, in case the writing has enough internal structure to warrant such markup. In either case the usual phrase-level tags for written text are available.

<xr> (cross-reference phrase) contains a phrase, sentence, or icon referring the reader to some other location in this or another text.

Module dictionaries — *9. Dictionaries*

Attributes att.lexicographic (*@expand*, *@norm*, *@split*, *@value*, *@orig*, *@location*, *@mergedIn*, *@opt*)

@type indicates the type of cross reference, using any convenient typology.

Status Recommended

Datatype `data.enumerated`

Sample values include: **syn** (synonym) cross reference for synonym information

etym (etymological) etymological information

cf (compare or consult) related or similar term

illus (illustration) illustration of an object

Used by etym model.entryPart.top model.entryPart

May contain

analysis: c cl interp interpGrp m phr s span spanGrp w

certainty: certainty respons

core: abbr add address bibl biblStruct binaryObject cb choice cit corr date del desc distinct email emph expan foreign gap gloss graphic hi index label lb list listBibl measure measureGrp mentioned milestone name note num orig pb ptr q quote ref reg rs said sic soCalled stage term time title unclear

dictionaries: lang lbl oRef oVar pRef pVar usg

drama: camera caption castList move sound tech view

figures: figure formula table

gaiji: g

header: biblFull

iso-fs: fLib fs fvLib

linking: alt altGrp anchor join joinGrp link linkGrp seg timeline

msdescription: catchwords depth dimensions height heraldry locus material msDesc origDate origPlace secFol signatures stamp watermark width

namesdates: addName affiliation bloc country district forename genName geo geogFeat geogName listNym listOrg listPerson listPlace nameLink offset orgName persName placeName region roleName settlement state surname

spoken: incident kinesic pause shift vocal writing

tagdocs: att classSpec code eg egXML elementSpec gi ident listRef macroSpec moduleRef moduleSpec specDesc specGrp specGrpRef specList tag val

textcrit: app listWit witDetail

transcr: addSpan am damage damageSpan delSpan ex fw handShift restore space subst supplied

verse: caesura rhyme

Declaration

```
element xr
{
   att.global.attributes,
   att.lexicographic.attributes,
   attribute type { data.enumerated }?,
   (
      text
    | model.gLike    | model.phrase    | model.inter    | usg    | lbl    | model.global   )*
}
```

Example

```
<entry>
  <form>
    <orth>lavage</orth>
  </form>
  <etym>[Fr. < <mentioned>laver</mentioned>;
     L. <mentioned>lavare</mentioned>,
     to wash; <xr>see <ref>lather</ref>
    </xr>].
  </etym>
</entry>
```

Example

```
<entry>
  <form>
    <orth>lawful</orth>
  </form>
  <xr  type="syn">SYN. see <ref>legal</ref>
  </xr>
</entry>
```

Note May contain character data and phrase-level elements; usually contains a <ref> or a <ptr> element.This element encloses both the actual indication of the location referred to, which may be tagged using the <ref> or <ptr> elements, and any accompanying material which gives more information about why the reader is being referred there.

<zone> defines a rectangular area contained within a <surface> element.

Module transcr — *11. Representation of Primary Sources*

Attributes att.coordinated (*@ulx*, *@uly*, *@lrx*, *@lry*)

Used by surface

May contain

core: binaryObject desc gloss graphic

figures: formula

tagdocs: altIdent equiv

Declaration

```
element zone
{
   att.global.attributes,
   att.coordinated.attributes,
   ( model.glossLike*, model.graphicLike* )
}
```

Example

```
<facsimile>
  <surface
     ulx="50"
     uly="20"
     lrx="400"
     lry="280">
    <zone
       ulx="0"
       uly="0"
       lrx="500"
       lry="321">
      <graphic  url="graphic.png "/>
    </zone>
  </surface>
</facsimile>
```

Note The position of every zone for a given surface is always defined by reference to the coordinate system defined for that surface. Any graphic element contained by a zone represents the whole of the zone.

Appendix D

Attributes

absolute	when
active	interaction relation
adj	node
adjFrom	node
adjTo	node
age	person personGrp
agent	att.damagedgap unclear
aloud	said
ana	att.global.analytic
anchored	note
arity	tree
assertedValue	certainty
atts	specDesc
baseForm	m
baseTypes	fsDecl
cRef	gloss ptr ref term
calendar	date
cause	att.textCritical
cert	att.editLike
children	iNode root
class	msContents msItem msItemStruct
code	occupation socecStatus
cols	att.tableDecorationtable
columns	layout
commodity	att.measurement
contemporary	binding seal
copyOf	att.global.linking
corresp	att.global.linking
datum	geoDecl
decls	att.declaring
default	att.declarable
defective	att.msExcerpt
degree	att.damagedcertainty node purpose
delim	refState

dim	space
direct	said
discrete	sound
docLang	schemaSpec
domains	att.pointing.group
dur	att.duration.w3c
dur-iso	att.duration.iso
ed	cb lb milestone pb refState
encoding	binaryObject
end	att.timed
enjamb	att.enjamb
eol	hyphenation
evaluate	att.pointing
evidence	att.editLike
exclude	att.global.linking
expand	att.lexicographic
extent	att.damagedgap orth pron
fVal	f
facs	att.global.facs
feats	fs
feature	shift
filter	equiv
follow	iNode leaf
form	objectDesc quotation
from	att.datable.w3capp arc locus span
from-iso	att.datable.iso
full	att.personal
function	att.segLike
generate	classSpec
gi	tagUsage
given	certainty
gradual	writing
group	att.damaged
hand	att.damagedatt.textCriticalatt.transcriptionalgap unclear
hands	handDesc
height	binaryObject graphic
ident	att.identifiedapplication language
inDegree	node
indexName	index
inst	att.interpLike
interval	timeline when
iterated	kinesic vocal
key	att.namingmemberOf moduleRef specDesc
label	rhyme
lang	code
lemma	w
lemmaRef	w
length	refState

level	langKnown sense title
loc	app
location	att.lexicographicvariantEncoding
locus	certainty respons
lrx	att.coordinated
lry	att.coordinated
mainLang	textLang
marks	quotation
matchPattern	cRefPattern
material	supportDesc
max	numeric
maxOccurs	datatype
medium	att.handFeatures
mergedIn	att.lexicographic
met	att.metrical
method	correction normalization variantEncoding
mimeType	att.internetMedia
minOccurs	datatype
mode	att.identifiedalt altGrp channel classes memberOf valDesc valList
module	att.identified
mutual	relation
n	att.global
name	attRef equiv f fDecl namespace relation vLabel
new	handShift shift
next	att.global.linking
norm	att.lexicographic
notAfter	att.datable.w3c
notAfter-iso	att.datable.iso
notBefore	att.datable.w3c
notBefore-iso	att.datable.iso
notation	formula pron
ns	attDef elementSpec schemaSpec
nymRef	att.naming
occurs	tagUsage
opt	att.lexicographic
optional	fDecl
ord	iNode root tree
order	graph tree
org	att.divLikeattList vColl vMerge
orig	att.lexicographic
origin	timeline
otherLangs	textLang
outDegree	iNode node root
parent	iNode leaf
part	att.divLikeatt.segLikeab l
parts	nym
passive	interaction relation
pattern	metDecl

perf	move tech
period	att.datable.w3c
place	att.placement
precision	att.dimensionsatt.editLike
predeclare	att.identified
prefix	schemaSpec
prev	att.global.linking
quantity	att.dimensionsatt.measurement
real	att.metrical
reason	gap supplied unclear
ref	att.namingg
rend	att.global
render	tagUsage
rendition	att.global
replacementPattern	cRefPattern
resp	att.editLikeatt.interpLikeatt.textCriticalhandShift note respons space witDetail
result	join joinGrp
rhyme	att.metrical
role	att.tableDecorationeditor org person personGrp
rows	att.tableDecorationtable
ruledLines	layout
sameAs	att.global.linking
sample	att.divLike
scale	binaryObject graphic
scheme	att catRef classCode gi keywords locus occupation rendition socecStatus tag
scope	att.dimensionsatt.handFeaturesjoin
scribe	att.handFeatures
script	att.handFeatures
select	att.global.linking
seq	att.transcriptional
sex	person personGrp
since	when
size	graph personGrp
social	distinct
sort	att.personal
sortKey	att.entryLiketerm
source	att.editLikenormalization writing
space	distinct
spanTo	att.spanning
split	att.lexicographic
start	att.timedschemaSpec surface
status	att.transcriptionalavailability correction
subtype	att.typedseg
synch	att.global.linking
tag	langKnown
tags	langKnowledge
targFunc	att.pointing.group

target	att.ptrLike.formcatRef certainty fsdLink gloss locus note ptr ref respons specGrpRef term witDetail
targetEnd	note
targetLang	schemaSpec
targets	alt join link
terminal	metSym
time	distinct
to	att.datable.w3capp arc locus span
to-iso	att.datable.iso
trans	u
trunc	numeric
type	att.entryLikeatt.interpLikeatt.pointingatt.segLikeatt.textCriticalatt.typedabbr app biblScope castItem classSpec constitution derivation dimensions distinct divGen domain factuality forest forestGrp form fs fsDecl fsdLink fw geogName gram graph iType idno interaction lbl list macroSpec measure metDecl moduleSpec move name node note num oRef oVar orth preparedness purpose q recording relation rs sound stage tag tech teiHeader title titlePage titlePart usg valList witDetail xr
ulx	att.coordinated
uly	att.coordinated
unit	att.dimensionsatt.measurementgap milestone refState timeline when
uri	equiv
url	graphic moduleRef
usage	attDef elementSpec language
value	att.lexicographicage binary eLeaf eTree iNode leaf metSym node num numeric root sex symbol triangle
varSeq	att.textCritical
version	att.translatableTEI application teiCorpus unicodeName
weights	alt
when	att.datable.w3cchange docDate
when-iso	att.datable.iso
where	event move
who	att.ascribed
width	binaryObject graphic
wit	att.rdgPartatt.textCriticalwitDetail
withId	tagUsage
writtenLines	layout
xml:base	att.global
xml:id	att.global
xml:lang	att.global
xml:space	att.xmlspace

Appendix E

Datatypes and Other Macros

---

data.certainty defines the range of attribute values expressing a degree of certainty.

Module tei — *1. The TEI Infrastructure*

Used by Class:

- att.dimensions/@precision
- att.editLike/@cert
- att.editLike/@precision

Element:

- purpose/@degree

Declaration

```
data.certainty = "high" | "medium" | "low" | "unknown"
```

Note Certainty may be expressed by one of the predefined symbolic values **high**, **medium**, or **low**. For more precise indication, data.probability may be used instead or in addition.

---

data.code defines the range of attribute values expressing a coded value by means of a pointer to some other element which contains a definition for it.

Module tei — *1. The TEI Infrastructure*

Used by Element:

- cb/@ed
- distinct/@time
- distinct/@space
- distinct/@social
- formula/@notation
- handShift/@new
- handShift/@resp
- lb/@ed
- milestone/@ed
- msContents/@class

- msItem/@class
- msItemStruct/@class
- pb/@ed
- refState/@ed
- writing/@source

Declaration `data.code = xsd:anyURI`

Note It will usually be the case that the item pointed to is to be found somewhere else in the current TEI document, typically in the header, but this is not mandatory.

---

data.count defines the range of attribute values used for a non-negative integer value used as a count.

Module tei — *1. The TEI Infrastructure*

Used by Class:

- att.damaged/@group
- att.tableDecoration/@rows
- att.tableDecoration/@cols
- att.textCritical/@varSeq
- att.transcriptional/@seq

Element:

- age/@value
- datatype/@minOccurs
- graph/@order
- graph/@size
- handDesc/@hands
- iNode/@outDegree
- layout/@columns
- layout/@ruledLines
- layout/@writtenLines
- node/@inDegree
- node/@outDegree
- node/@degree
- refState/@length
- root/@outDegree
- table/@rows
- table/@cols
- tagUsage/@occurs
- tagUsage/@withId
- tree/@arity

- tree/@order

Declaration `data.count = xsd:nonNegativeInteger`

Note Only positive integer values are permitted

---

data.duration.iso defines the range of attribute values available for representation of a duration in time using ISO 8601 standard formats

Module tei — *1. The TEI Infrastructure*

Used by Class:

- att.duration.iso/@dur-iso

Declaration

```
data.duration.iso = token { pattern = "[0-9.,DHMPRSTWYZ/:+\-]+" }
```

Example

```
<time dur-iso="PT0,75H">three-quarters of an hour</time>
```

Example

```
<date dur-iso="P1,5D">a day and a half</date>
```

Example

```
<date dur-iso="P14D">a fortnight</date>
```

Example

```
<time dur-iso="PT0.02S">20 ms</time>
```

Note A duration is expressed as a sequence of number-letter pairs, preceded by the letter P; the letter gives the unit and may be Y (year), M (month), D (day), H (hour), M (minute), or S (second), in that order. The numbers are all unsigned integers, except for the last, which may have a decimal component (using either `.` or `,` as the decimal point; the latter is preferred). If any number is *0*, then that number-letter pair may be omitted. If any of the H (hour), M (minute), or S (second) number-letter pairs are present, then the separator `T` must precede the first 'time' number-letter pair.For complete details, see ISO 8601 *Data elements and interchange formats — Information interchange — Representation of dates and times.*

---

data.duration.w3c defines the range of attribute values available for representation of a duration in time using W3C datatypes.

Module tei — *1. The TEI Infrastructure*

Used by Class:

- att.duration.w3c/@dur

Declaration `data.duration.w3c = xsd:duration`

Example

```
<time dur="PT45M">forty-five minutes</time>
```

Example

```
<date dur="P1DT12H">a day and a half</date>
```

Example

```
<date dur="P7D">a week</date>
```

Example

```
<time dur="PT0.02S">20 ms</time>
```

Note A duration is expressed as a sequence of number-letter pairs, preceded by the letter P; the letter gives the unit and may be Y (year), M (month), D (day), H (hour), M (minute), or S (second), in that order. The numbers are all unsigned integers, except for the `S` number, which may have a decimal component (using `.` as the decimal point). If any number is *0*, then that number-letter pair may be omitted. If any of the H (hour), M (minute), or S (second) number-letter pairs are present, then the separator `T` must precede the first 'time' number-letter pair.For complete details, see the W3C specification.

---

data.enumerated defines the range of attribute values expressed as a single XML name taken from a list of documented possibilities.

Module tei — *1. The TEI Infrastructure*

Used by Class:

- att.damaged/@agent
- att.dimensions/@unit
- att.dimensions/@scope
- att.editLike/@evidence
- att.enjamb/@enjamb
- att.entryLike/@type
- att.handFeatures/@medium
- att.interpLike/@type
- att.measurement/@unit
- att.placement/@place
- att.pointing/@type
- att.segLike/@type
- att.segLike/@function
- att.tableDecoration/@role
- att.textCritical/@type
- att.textCritical/@cause
- att.transcriptional/@status

- att.typed/@type
- att.typed/@subtype

Element:

- abbr/@type
- app/@type
- att/@scheme
- biblScope/@type
- certainty/@locus
- classSpec/@generate
- date/@calendar
- derivation/@type
- dimensions/@type
- distinct/@type
- divGen/@type
- domain/@type
- editor/@role
- forest/@type
- forestGrp/@type
- form/@type
- fs/@type
- fsDecl/@type
- fsdLink/@type
- fw/@type
- gap/@agent
- gap/@unit
- geoDecl/@datum
- geogName/@type
- gi/@scheme
- gram/@type
- graph/@type
- iType/@type
- idno/@type
- interaction/@active
- interaction/@passive
- lbl/@type
- list/@type
- measure/@type

- metDecl/@type
- milestone/@unit
- move/@type
- move/@where
- name/@type
- node/@type
- note/@type
- num/@type
- oRef/@type
- oVar/@type
- objectDesc/@form
- orth/@type
- orth/@extent
- person/@role
- person/@age
- personGrp/@role
- personGrp/@age
- preparedness/@type
- pron/@extent
- pron/@notation
- purpose/@type
- q/@type
- refState/@unit
- relation/@type
- relation/@name
- respons/@locus
- rs/@type
- shift/@new
- sound/@type
- stage/@type
- supportDesc/@material
- tech/@perf
- teiHeader/@type
- timeline/@unit
- title/@type
- titlePage/@type
- titlePart/@type

- unclear/@agent
- usg/@type
- when/@unit
- witDetail/@type
- xr/@type

Declaration `data.enumerated = data.name`

Note Attributes using this datatype must contain a word which follows the rules defining a legal XML name (see `http://www.w3.org/TR/REC-xml/#dt-name`): for example they cannot include whitespace or begin with digits. Typically, the list of documented possibilities will be provided (or exemplified) by a value list in the associated attribute specification, expressed with a <valList> element.

---

data.key defines the range of attribute values expressing a coded value by means of an arbitrary identifier, typically taken from a set of externally-defined possibilities.

Module tei — *1. The TEI Infrastructure*

Used by Class:

- att.naming/@key

Element:

- w/@lemma

Declaration `data.key = string`

Note Information about the set of possible values for an attribute using this datatype may (but need not) be documented in the document header. Externally defined constraints, for example that values should be legal keys in an external database system, cannot usually be enforced by a TEI system. Similarly, because the key is externally defined, no constraint other than a requirement that it consist of Unicode characters is possible.

---

data.language defines the range of attribute values used to identify a particular combination of human language and writing system.

Module tei — *1. The TEI Infrastructure*

Used by Class:

- att.global/@xml:lang

Element:

- langKnowledge/@tags
- langKnown/@tag
- language/@ident
- schemaSpec/@targetLang
- schemaSpec/@docLang
- textLang/@mainLang
- textLang/@otherLangs

Declaration `data.language = xsd:language`

Note The values for this attribute are language 'tags' as defined in BCP 47. Currently BCP 47 comprises RFC 4646 and RFC 4647; over time, other IETF documents may succeed these as the best current practice.A 'language tag', per BCP 47, is assembled from a sequence of components or subtags separated by the hyphen character (-, U+002D). The tag is made of the following subtags, in the following order. Every subtag except the first is optional. If present, each occurs only once, except the fourth and fifth components (variant and extension), which are repeatable.

language The IANA-registered code for the language. This is almost always the same as the ISO 639 2-letter language code if there is one. The list of available registered language subtags can be found at `http://www.iana.org/assignments/language-subtag-registry`. It is recommended that this code be written in lower case.

script The ISO 15924 code for the script. These codes consist of 4 letters, and it is recommended they be written with an initial capital, the other three letters in lower case. The canonical list of codes is maintained by the Unicode Consortium, and is available at `http://unicode.org/iso15924/iso15924-codes.html`. The IETF recommends this code be omitted unless it is necessary to make a distinction you need.

region Either an ISO 3166 country code or a UN M.49 region code that is registered with IANA (not all such codes are registered, e.g. UN codes for economic groupings or codes for countries for which there is already an ISO 3166 2-letter code are not registered). The former consist of 2 letters, and it is recommended they be written in upper case. The list of codes can be found at `http://www.iso.org/iso/en/prods-services/iso3166ma/02iso-3166-code-lists/index.html`. The latter consist of 3 digits; the list of codes can be found at `http://unstats.un.org/unsd/methods/m49/m49.htm`.

variant An IANA-registered variation. These codes 'are used to indicate additional, well-recognized variations that define a language or its dialects that are not covered by other available subtags'.

extension An extension has the format of a single letter followed by a hyphen followed by additional subtags. These exist to allow for future extension to BCP 47, but as of this writing no such extensions are in use.

private use An extension that uses the initial subtag of the single letter *x* (i.e., starts with `x-`) has no meaning except as negotiated among the parties involved. These should be used with great care, since they interfere with the interoperability that use of RFC 4646 is intended to promote. In order for a document that makes use of these subtags to be TEI conformant, a corresponding <language> element must be present in the TEI header.

There are two exceptions to the above format. First, there are language tags in the IANA registry that do not match the above syntax, but are present because they have been 'grandfathered' from previous specifications.Second, an entire language tag can consist of only a private use subtag. These tags start with `x-`, and do not need to follow any further rules established by the IETF and endorsed by these Guidelines. Like all language tags that make use of private use subtags, the language in question must be documented in a corresponding <language> element in the TEI header.Examples include

sn Shona

zh-TW Taiwanese

zh-Hant-HK Chinese written in traditional script as used in Hong Kong

en-SL English as spoken in Sierra Leone

pl Polish

es-MX Spanish as spoken in Mexico

es-419 Spanish as spoken in Latin America

The W3C Internationalization Activity has published a useful introduction to BCP 47, Language tags in HTML and XML.

---

data.name defines the range of attribute values expressed as an XML Name.

Module tei — *1. The TEI Infrastructure*

Used by Class:

- att.handFeatures/@scribe
- att.handFeatures/@script
- att.identified/@ident
- att.pointing.group/@domains

Element:

- application/@ident
- equiv/@name
- f/@name
- fDecl/@name
- fsDecl/@baseTypes
- index/@indexName
- join/@result
- joinGrp/@result
- memberOf/@key
- schemaSpec/@start
- schemaSpec/@prefix
- specDesc/@key
- specDesc/@atts
- tagUsage/@gi

data.enumerated

Declaration `data.name = xsd:Name`

Note Attributes using this datatype must contain a single word which follows the rules defining a legal XML name (see `http://www.w3.org/TR/REC-xml/#dt-name`): for example they cannot include whitespace or begin with digits.

---

data.namespace defines the range of attribute values used to indicate XML namespaces as defined by the W3C Namespaces in XML Technical Recommendation.

Module tei — *1. The TEI Infrastructure*

Used by Element:

- attDef/@ns
- elementSpec/@ns
- namespace/@name
- schemaSpec/@ns

Declaration `data.namespace = xsd:anyURI`

Note The range of syntactically valid values is defined by RFC 3986 *Uniform Resource Identifier (URI): Generic Syntax*

---

data.numeric defines the range of attribute values used for numeric values.

Module tei — *1. The TEI Infrastructure*

Used by Class:

- att.coordinated/@ulx
- att.coordinated/@uly
- att.coordinated/@lrx
- att.coordinated/@lry
- att.dimensions/@quantity
- att.measurement/@quantity

Element:

- binaryObject/@scale
- gap/@extent
- graphic/@scale
- num/@value
- numeric/@value
- numeric/@max
- sense/@level
- unicodeName/@version

Declaration `data.numeric = xsd:double | xsd:decimal`

Note Any numeric value that can be represented as a decimal number.In addition, the range of values that can be represented in an IEEE double precision (i.e., 64-bit) floating point number may be represented using scientific notation. Roughly that range is $\pm 10^{-323}$ to $\pm 10^{308}$.To represent a number expressed in scientific notation, 'E notation', a variant of 'exponential notation', is used in the attribute value. A number is represented in exponential notation as a value between 1 and 10 (or -1 and -10) which is multiplied by ten raised to some integral power. When reading E notation, the letter *E* is read as 'times 10 to the power'.That is, the significand (sometimes called the mantissa) is written as a decimal number, followed by the letter *E*, followed by an integer exponent. The multiplication sign and the base itself (*10*) are implied. Either the significand or the exponent (or both) may be a negative number, in which case it should be preceded by a minus

sign. There should be no whitespace separating the significand from the *E* from the exponent. E.g., 3×10^8 can be expressed as **3E8**.Other examples of scientific notation include:

- **3E10** (the speed of light in centimetres per second)
- **9.12E-31** (the mass of an electron in Kg)
- **4E11** (estimated number of stars in our galaxy)
- **-1.76E11** (electron charge to mass quotient in coulombs per Kg)

Either *e* or *E* may be used to separate the significand from the exponent, however these Guidelines recommend *E* be used both for consistency with other standards bodies and to avoid confusion with the mathematical constant e.

---

data.outputMeasurement defines a range of values for use in specifying the size of an object that is intended for display on the web.

Module tei — *1. The TEI Infrastructure*

Used by Element:

- binaryObject/@width
- binaryObject/@height
- graphic/@width
- graphic/@height

Declaration

```
data.outputMeasurement =
   token
   {
      pattern = "[\-+]?\d+(\.\d+)?(%|cm|mm|in|pt|pc|px|em|ex|gd|rem|vw|vh|vm)"
   }
```

Example

```
<figure>
  <head>The TEI Logo</head>
  <figDesc>Stylized yellow angle brackets with the letters
  <mentioned>TEI</mentioned> in between and <mentioned>text
        encoding initiative</mentioned> underneath, all on a white
     background.</figDesc>
  <graphic
     height="600px"
     width="600px"
     url="http://www.tei-c.org/logos/TEI-600.jpg"/>
</figure>
```

Note These values map directly onto the values used by XSL-FO and CSS. For definitions of the units see those specifications; at the time of this writing the most complete list is in the CSS3 working draft.

---

data.pattern (regular expression pattern) defines attribute values which are expressed as a regular expression.

Module tei — *1. The TEI Infrastructure*

Used by Element:

- cRefPattern/@matchPattern
- metDecl/@pattern

Declaration `data.pattern = token`

Note `A regular expression, often called a pattern, is an expression that describes a set of strings. They are usually used to give a concise description of a set, without having to list all elements. For example, the set containing the three strings *Handel*, *Händel*, and *Haendel* can be described by the pattern `H(ä|ae?)ndel` (or alternatively, it is said that the pattern `H(ä|ae?)ndel`matches each of the three strings)'wikipedia

---

data.pointer defines the range of attribute values used to provide a single URI pointer to any other resource, either within the current document or elsewhere.

Module tei — *1. The TEI Infrastructure*

Used by Class:

- att.ascribed/@who
- att.damaged/@hand
- att.datable.w3c/@period
- att.declaring/@decls
- att.editLike/@resp
- att.editLike/@source
- att.global/@rendition
- att.global/@xml:base
- att.global.analytic/@ana
- att.global.facs/@facs
- att.global.linking/@corresp
- att.global.linking/@synch
- att.global.linking/@sameAs
- att.global.linking/@copyOf
- att.global.linking/@next
- att.global.linking/@prev
- att.global.linking/@exclude
- att.global.linking/@select
- att.interpLike/@resp
- att.interpLike/@inst
- att.lexicographic/@location
- att.lexicographic/@mergedIn
- att.naming/@ref

- att.naming/@nymRef
- att.ptrLike.form/@target
- att.rdgPart/@wit
- att.spanning/@spanTo
- att.textCritical/@wit
- att.textCritical/@resp
- att.textCritical/@hand
- att.timed/@start
- att.timed/@end
- att.transcriptional/@hand

Element:

- alt/@targets
- app/@from
- app/@to
- arc/@from
- arc/@to
- catRef/@target
- catRef/@scheme
- certainty/@target
- certainty/@given
- classCode/@scheme
- eLeaf/@value
- eTree/@value
- equiv/@uri
- equiv/@filter
- event/@where
- f/@fVal
- fs/@feats
- fsdLink/@target
- g/@ref
- gap/@hand
- gloss/@target
- gloss/@cRef
- graphic/@url
- iNode/@value
- iNode/@children
- iNode/@parent

- iNode/@follow
- join/@targets
- keywords/@scheme
- leaf/@value
- leaf/@parent
- leaf/@follow
- link/@targets
- locus/@scheme
- locus/@target
- moduleRef/@url
- move/@perf
- node/@value
- node/@adjTo
- node/@adjFrom
- node/@adj
- normalization/@source
- note/@resp
- note/@target
- note/@targetEnd
- nym/@parts
- occupation/@scheme
- occupation/@code
- ptr/@target
- ref/@target
- relation/@active
- relation/@mutual
- relation/@passive
- respons/@target
- respons/@resp
- root/@value
- root/@children
- socecStatus/@scheme
- socecStatus/@code
- space/@resp
- span/@from
- span/@to
- specGrpRef/@target

- surface/@start
- tagUsage/@render
- term/@target
- term/@cRef
- timeline/@origin
- triangle/@value
- unclear/@hand
- w/@lemmaRef
- when/@since
- witDetail/@target
- witDetail/@resp
- witDetail/@wit

Declaration `data.pointer = xsd:anyURI`

Note The range of syntactically valid values is defined by RFC 3986 *Uniform Resource Identifier (URI): Generic Syntax*

---

data.probability defines the range of attribute values expressing a probability.

Module tei — *1. The TEI Infrastructure*

Used by Element:

- alt/@weights
- certainty/@degree

Declaration

```
data.probability = xsd:double { minInclusive = "0" maxInclusive = "1" }
```

Note Probability is expressed as a real number between 0 and 1; 0 representing *certainly false* and 1 representing *certainly true*.

---

data.sex defines the range of attribute values used to identify human or animal sex.

Module tei — *1. The TEI Infrastructure*

Used by Element:

- person/@sex
- sex/@value

Declaration `data.sex = "0" | "1" | "2" | "9"`

Note The values are taken from ISO 5218:1977 *Representation of Human Sexes*; 0 indicates unknown; 1 indicates male; 2 indicates female; and 9 indicates not applicable.

---

data.temporal.iso defines the range of attribute values expressing a temporal expression such as a date, a time, or a combination of them, that conform to the international standard *Data elements and interchange formats – Information interchange – Representation of dates and times.*

Module tei — *1. The TEI Infrastructure*

Used by Class:

- att.datable.iso/@when-iso
- att.datable.iso/@notBefore-iso
- att.datable.iso/@notAfter-iso
- att.datable.iso/@from-iso
- att.datable.iso/@to-iso

Declaration

```
data.temporal.iso =
   xsd:date
 | xsd:gYear
 | xsd:gMonth
 | xsd:gDay
 | xsd:gYearMonth
 | xsd:gMonthDay
 | xsd:time
 | xsd:dateTime
 | token { pattern = "[0-9.,DHMPRSTWYZ/:+\-]+" }
```

Note If it is likely that the value used is to be compared with another, then a time zone indicator should always be included, and only the dateTime representation should be used.For all representations for which ISO 8601 describes both a basic and an extended format, these Guidelines recommend use of the extended format.While ISO 8601 permits the use of both `00:00` and `24:00` to represent midnight, these Guidelines strongly recommend against the use of `24:00`.

---

data.temporal.w3c defines the range of attribute values expressing a temporal expression such as a date, a time, or a combination of them, that conform to the W3C *XML Schema Part 2: Datatypes* specification.

Module tei — *1. The TEI Infrastructure*

Used by Class:

- att.datable.w3c/@when
- att.datable.w3c/@notBefore
- att.datable.w3c/@notAfter
- att.datable.w3c/@from
- att.datable.w3c/@to

Element:

- change/@when
- docDate/@when
- when/@absolute

Declaration

```
data.temporal.w3c =
   xsd:date
 | xsd:gYear
 | xsd:gMonth
 | xsd:gDay
 | xsd:gYearMonth
 | xsd:gMonthDay
 | xsd:time
 | xsd:dateTime
```

Note If it is likely that the value used is to be compared with another, then a time zone indicator should always be included, and only the dateTime representation should be used.

---

data.truthValue defines the range of attribute values used to express a truth value.

Module tei — *1. The TEI Infrastructure*

Used by Class:

- att.declarable/@default

Element:

- binary/@value
- metSym/@terminal
- note/@anchored
- numeric/@trunc

Declaration `data.truthValue = xsd:boolean`

Note This datatype applies only for cases where uncertainty is inappropriate; if the attribute concerned may have a value other than true or false, e.g. "unknown", or "inapplicable", it should have the extended version of this datatype: data.xTruthValue

---

data.word defines the range of attribute values expressed as a single word or token.

Module tei — *1. The TEI Infrastructure*

Used by Class:

- att.damaged/@extent
- att.entryLike/@sortKey
- att.global/@n
- att.global/@rend
- att.internetMedia/@mimeType
- att.measurement/@commodity
- att.pointing.group/@targFunc
- att.translatable/@version

Element:

- app/@loc

- attRef/@name
- binaryObject/@encoding
- code/@lang
- gap/@reason
- langKnown/@level
- locus/@from
- locus/@to
- m/@baseForm
- metSym/@value
- org/@role
- personGrp/@size
- ptr/@cRef
- ref/@cRef
- rhyme/@label
- seg/@subtype
- supplied/@reason
- symbol/@value
- term/@sortKey
- unclear/@reason
- vLabel/@name

Declaration

```
data.word = token { pattern = "(\p{L}|\p{N}|\p{P}|\p{S})+" }
```

Note Attributes using this datatype must contain a single 'word' which contains only letters, digits, punctuation characters, or symbols: thus it cannot include whitespace.

---

data.xTruthValue (extended truth value) defines the range of attribute values used to express a truth value which may be unknown.

Module tei — *1. The TEI Infrastructure*

Used by Class:

- att.msExcerpt/@defective

Element:

- binding/@contemporary
- iNode/@ord
- kinesic/@iterated
- root/@ord
- said/@aloud
- said/@direct

- seal/@contemporary
- sound/@discrete
- vocal/@iterated
- writing/@gradual

Declaration

```
data.xTruthValue = xsd:boolean | "unknown" | "inapplicable"
```

Note In cases where where uncertainty is inappropriate, use the datatype data.TruthValue

---

macro.limitedContent (paragraph content) defines the content of prose elements that are not used for transcription of extant materials.

Module tei — *1. The TEI Infrastructure*

Used by desc fDescr figDesc fsDescr meeting rendition tagUsage witness

Declaration

```
macro.limitedContent = ( text | model.limitedPhrase | model.inter )*
```

---

macro.paraContent (paragraph content) defines the content of paragraphs and similar elements.

Module tei — *1. The TEI Infrastructure*

Used by ab add camera caption case cell colloc corr damage def del docEdition emph gen gram head hi hyph iType imprimatur l lang lbl mood number orig orth p per pos pron ref reg restore rhyme seg sic sound stress subc supplied syll tech title titlePart tns unclear usg writing

Declaration

```
macro.paraContent =
   ( text | model.gLike | model.phrase | model.inter | model.global )*
```

---

macro.phraseSeq (phrase sequence) defines a sequence of character data and phrase-level elements.

Module tei — *1. The TEI Infrastructure*

Used by abbr actor addName addrLine affiliation author biblScope birth bloc catchwords cl colophon country dateline death distinct distributor district docAuthor docDate edition editor education email expan explicit extent faith finalRubric floruit foreign forename fw genName geoDecl geogName gloss headItem headLabel heraldry incipit label material measure mentioned name nameLink nationality num occupation orgName persName phr placeName pubPlace publisher region residence role roleDesc roleName rs rubric s salute secFol settlement sex signatures signed soCalled socecStatus speaker stamp street summary surname term textLang trailer watermark wit witDetail

Declaration

```
macro.phraseSeq = ( text | model.gLike | model.phrase | model.global )*
```

macro.phraseSeq.limited (limited phrase sequence) defines a sequence of character data and those phrase-level elements that are not typically used for transcribing extant documents.

Module tei — *1. The TEI Infrastructure*

Used by activity age authority channel classCode constitution creation derivation domain factuality funder interaction langKnown language locale metSym preparedness principal purpose resp span sponsor valDesc

Declaration

```
macro.phraseSeq.limited = ( text | model.limitedPhrase | model.global )*
```

macro.schemaPattern provides a pattern to match elements from the chosen schema language

Module tagdocs — *22. Documentation Elements*

Used by content datatype

Declaration

```
macro.schemaPattern = text
```

macro.specialPara ('special' paragraph content) defines the content model of elements such as notes or list items, which either contain a series of component-level elements or else have the same structure as a paragraph, containing a series of phrase-level and inter-level elements.

Module tei — *1. The TEI Infrastructure*

Used by accMat acquisition additions collation condition custEvent decoNote filiation foliation handNote item layout musicNotation note origin provenance q quote said source stage support surrogates view

Declaration

```
macro.specialPara =
   (
      text
    | model.gLike    | model.phrase    | model.inter    | model.divPart    | model.global   )*
```

macro.xtext (extended text) defines a sequence of character data and gaiji elements.

Module tei — *1. The TEI Infrastructure*

Used by altIdent am c collection depth ex geogFeat height institution locus mapping memberOf msName offset origPlace repository string value width

Declaration

```
macro.xtext = ( text | model.gLike )*
```

Appendix F

Bibliography

Works cited in examples in the Guidelines

[1] Adams, Douglas. *The Hitchhiker's Guide to the Galaxy* (1979), chapter 31.

[2] Alighieri, Dante. `Doglia mi reca ne lo core ardire', *Rime*, XLIX.

[3] Allinson, E.P. and B. Penrose. *Philadelphia 1681-1887* (1887), p.138.

[4] *American National Standard for Bibliographic References, ANSI Z39.29-1977*, New York: American National Standards Institute (1977)

[5] Atkins et al. *Collins Robert French-English English-French Dictionary*. London: Collins (1978)

[6] Atkinson, J. Maxwell and John Heritage. *Structures of social action: Studies in conversation analysis*, Cambridge and Paris: Cambridge University Press, Editions de la Maison des Sciences de l'Homme (1984), ix-xvi.

[7] Austen, Jane. *Pride and Prejudice*. (1813), chapter 1.

[8] Barbauld, Lucy Aikin. *The Works of Anna Laetitia Barbauld*. (1826)

[9] Barker, Jane. *The Lining to the Patch-Work Screen*. (1726)

[10] Beckett, Samuel. *Waiting for Godot*, London: Faber and Faber (1956)

[11] Beckett, Samuel. *Murphy* (1963), chap 2.

[12] Beerbohm, Max. Autograph manuscript of *The Golden Drugget*, Pierpont Morgan MA 3391. in Klinkenborg and Cahoon (1981) 123.

[13] Behn, Aphra. *The Rover*, (1697).

[14] *Beowulf and The fight at Finnsburg*; edited, with introduction, bibliography, notes, glossary, and appendices, by Fr. Klaeber. Boston, New York [etc.] D.C. Heath & Co. (1922)

[15] Wrenn C. L. ed. *Beowulf: with the Finnesburg fragment*, London: Harrap (1953)

[16] Blake, William. `London', in *Songs of Experience* (1791)

[17] Bloomfield, Leonard. `Literate and Illiterate Speech', *American Speech*, 2 , (1927), pp. 432-441.

[18] Borges, Jorge Luis, tr. R. Simms. `The Analytical Language of John Wilkins'. In Emir Rodriguez Monegal and Alistair Reid, eds. *Borges: A reader*, Dutton Adult (1981), p.141.

[19] Borges, Jorge Luis. `Avatars of the Tortoise' In James E. Irby tr. *Labyrinths: Selected Stories and Other Writings*, New York: New Directions, (1962), pp.202-203.

[20] Extract from *British National Corpus* (`http://www.natcorp.ox.ac.uk`) Text KB7, sentence 13730.

[21] Browning, Robert. *Letter to George Moulton-Barrett*, Pierpont Morgan MA 310, (Klinkenborg and Cahoon (1981) 23)

[22] Bunyan, John. *The Pilgrim's Progress from this world to that which is to come...*, London (1678)

[23] Burgess, Anthony. *A Clockwork Orange*. (1962), opening.

[24] Burton, Robert. *Anatomy of Melancholy* (1621), 16th ed. reprinted 1846, p.743.

[25] Butler, Samuel. *The Way of All Flesh* (1903), chapter 37.

[26] Byron, George Gordon. *Don Juan* (1819), I.xxii.

[27] Byron, George Gordon. `Vision of Judgment' In E.H. Coleridge ed. *The Poetical Works of Lord Byron*, viii, 1922.

[28] *C 60/16 Fine Roll 6 HENRY III (28 October 1221-27 October 1222)*, membrane 5, entry 154.

[29] Edward Barkley, describing how Essex drove the Irish from the plains into the woods to freeze or famish in winter; quoted by Canny, Nicholas P.`The Ideology of English Colonization: From Ireland to America'. In Stanley N. Katz and John M. Murrin eds. *Colonial America: Essays in Politics and Social Development*, 3d edNew York: Knopf, (1983), p.53.

[30] Carroll, Lewis. *Through the Looking Glass, and what Alice found there.* (1871)

[31] `The Castle of the Fly', in *Russian Fairy Tales,* translated by Norbert Guterman from the collections of Aleksandr Afanas'ev, illustrations by Alexander Alexeieff, folkloristic commentary by Roman Jakobson (New York: Pantheon Books, 1947, rpt. [n.d.]), p. 25.

[32] Example recoded from Chafe, W. `Adequacy, user-friendliness, and practicality in transcribing' In Leech, G., G. Myers, J. Thomas eds. *Spoken English on Computer: Transcription, Markup and Applications.* Harlow: Longman, 1995.

[33] Chandler, Lloyd.`Conversation with Death' (also known as `Oh, Death'). In *Journal of Folklore Research*, 41.2/3 , (2004), pp. 125-126

[34] Chaucer, Geoffrey. *Canterbury Tales*, f52r, in Holkham MS

[35] Chaucer, Geoffrey. `The Tale of Sir Topas', *The Canterbury Tales*, In F. N. Robinson ed. *The Works of Geoffrey Chaucer*, 2nd editionBoston: Houghton Mifflin Co., 1957.

[36] Chomsky, Noam and Morris Halle. *The Sound Pattern of English*. New York: Harper & Row (1968), p. 415.

[37] Cleaver, Eldridge. *Soul on Ice*. New York (1968)

[38] `Cloud of Unknowing' In Hodgson, Phyllis ed. *The Cloud of Unknowing and The Book of Privy Counselling*, London: Oxford University Press, Early English Text Society, 218, (1944)

[39] Clover, Carol J.*The Medieval Saga*, Ithaca: Cornell University Press (1982)

[40] Cocteau, Jean. *La Machine Infernale*.

[41] Comenius, John Amos. *Orbis Pictus: a facsimile of the first English edition of 1659* (ed. John E. Sadler) Oxford University Press (1968)

[42] Coleridge, Samuel Taylor. *The Rime of the Ancient Mariner*. In Wordsworth, William and Samuel Taylor Coleridge. *Lyrical Ballads* (1798)

[43] Coleridge, Samuel Taylor. `Frost at Midnight' In E.H. Coleridge ed. *Poetical Works*, Oxford: Oxford University Press, (1967), p.240.

[44] Sinclair, John ed. *Collins COBUILD English Language Dictionary*. London and Glasgow: Collins, (1987), p.337 s.v. croissant.

[45] *Collins English Dictionary*. London: Collins

[46] *Collins Pocket Dictionary of the English language*. London: Collins

[47] Collins, Wilkie. *The Moonstone*, Penguin, 6th narrative.

[48] Cope, Thomas Pym. *Philadelphia merchant: the diary of Thomas P. Cope, 1800-1851*, Eliza Cope Harrison ed.

[49] Crashaw, Richard, ed. J.R. Tutin. *The Poems of Richard Crashaw*. Muses Library Edition: (1900)

[50] *The Daily Telegraph*21 Dec 1992.

[51] Dallas, George Mifflin. *Unpublished letter* cited in Russell F. Weigley, Nicholas B. Wainwright, Edwin Wolf eds. *Philadelphia: A 300 Year History*, New York and London: W. W. Norton & Company, 1982, p.349.

[52] Dean of Sarum Churchwardens' presentments, 1731, Hurst; Wiltshire Record Office; transcribed by Donald A. Spaeth.

[53] *De Nutrimento et Nutribili, Tractatus 1*, fol 217r col b of Merton College Oxford MS O.2.1; in Parkes (1969) pl. 16.

[54] Defoe, Daniel. *Robinson Crusoe* (1719).

[55] Defoe, Daniel. *Journal of the Plague Year*. London (1722)

[56] Dekker, Thomas and Thomas Middleton. *The Honest Whore, Part One* (1604)

[57] *Des Minnesangs Frühling*, Moser, Hugo, Helmut Tervooren eds. 36., neugestaltete und erweiterte AuflageI Texte, Stuttgart: S. Hirzel Verlag, 1977.

[58] Dickens, Charles. *A Christmas Carol in Prose, Being a Ghost Story of Christmas*, Chapman and Hall, (1843), p.12.

[59] Dickens, Charles. *Little Dorrit*, (1857).

[60] Dickinson, Emily. `1755' In Arthur Eastman et al. eds. *The Norton Anthology of Poetry*, New York: W.W. Norton, 1970, p.859.

[61] Disraeli, Benjamin. *Coningsby* (1844), preface.

[62] Doyle, Arthur Conan. `The Red-headed league'. In *The Adventures of Sherlock Holmes*. (1892)

[63] Doyle, Arthur Conan. *The Original Illustrated Sherlock Holmes*, Castle Books, 1989.

[64] Dudo of St Quentin. *De moribus et actis primorum Normannie ducum*, fol 4v of British Library MS Harley 3742; in Parkes (1969) pl 6(i).

[65] Dylan, Bob. `All Along the Watchtower' In *John Wesley Harding* (1967)

[66] *Eddic poems*, in Reykjavík, Landsbókasafn Íslands, Lbs 1562 4to

[67] `Editorial', *.EXE magazine*6.11 (1992), p.2.

[68] Eliot, George. *Middlemarch* (1871), chapter 1.

[69] Eliot, George. *Daniel Deronda* (1876), III.1.

[70] Eliot, Thomas Stearns. *The waste land: a facsimile and transcript of the original drafts including the annotations of Ezra Pound*, Eliot, Valerie ed. Faber and Faber Ltd. (1971), p.37.

[71] Fielding, Henry. `The History of the Adventures of Joseph Andrews and his Friend, Mr. Abraham Abrams' (1742).

[72] Fielding, Henry. *Tragedy of Tragedies*, (1737).

[73] Fish, Stanley. *Is there a text in this class? The authority of interpretive communities.* Harvard University Press (1980)

[74] Fisher, M. F. K. `I Was Really Very Hungry' In *As They Were*, Knopf (1982), p. 43.

[75] Foley James D., Andries van Dam, Steven K. Feiner and John F. Hughes. *Computer Graphics: Principles and Practice*, 2nd edition Reading: Addison-Wesley, p.259.

[76] Fussell, Paul. *The Norton Book of Travel*. W. W. Norton (1987)

[77] Galilei, Galileo, *Sidereus Nuncius*, Venetiis: Apud Thomam Baglionum, 1610, quoted by Tufte, Edward R., *Envisioning Information*, Cheshire: Graphics Press (1990), p.97.

[78] Gaskell, Elizabeth Cleghorn. *The Grey Woman*, MS.

[79] Gavioli, Laura and Gillian Mansfield. *The PIXI corpora: bookshop encounters in English and Italian*, Bologna: Cooperativa Libraria Universitaria Editrice (1990), p.74.

[80] Gay, John. *The Beggar's Opera* (1728).

[81] Gazdar, Gerald and Mellish, Christopher. *Natural language processing in Prolog*. Addison-Wesley (1989), p.5.

[82] Gazdar, Gerald, Ewan Klein, Geoffrey Pullum, and Ivan Sag: *Generalized Phrase Structure Grammar*, Harvard University Press (1985)

[83] *The Holy Bible, conteyning the Old Testament and the new... appointed to be read in Churches.* (1611), Genesis 1:1.

[84] Gibbon, Edward, *The History of the Decline and Fall of the Roman Empire*, (1789), chapter 58.

[85] Gilbert, William Schwenck and Sullivan, Arthur. *HMS Pinafore* (1878), I.

[86] Gilbert, William Schwenck and Sullivan, Arthur. *The Mikado* (1885).

[87] Ginsberg, Allen. `My alba', *Reality Sandwiches*, San Francisco: City Lights, (1963).

[88] Goethe, Johann Wolfgang von, tr. Philip Wayne. *Faust*, Part 1, London: Penguin. (1949)

[89] Goethe, Johann Wolfgang von. *Auf dem See*. (1775)

[90] Graves, Robert. *Rough draft of letter to Desmond Flower*. 17 Dec 1938. (from *Diary of Robert Graves 1935-39 and ancillary materials*, compiled by Beryl Graves, C.G. Petter, L.R. Roberts, University of Victoria Libraries)

[91] Greene, Robert. *Groatsworth of Wit Bought with a Million of Repentance* (1592).

[92] Gregory of Tours. *Ecclesiastical History of the Franks*

[93] Grune, Dick and Ceriel J. H. Jacobs. *Parsing Techniques: A Practical Guide*, New York and London: Ellis Horwood, 1990, p.24.

[94] Guerard, Françoise. *Le Dictionnaire de Notre Temps*, ed. Paris: Hachette, 1990

[95] Halliday, M.A.K. and R. Hassan. *Language, Context and Text: Aspects of Language in a Social-Semiotic Perspective*, Oxford: Oxford University Press, 1990, p.104.

[96] Hanks, Patrick. `Definitions and Explanations'. In J. M. Sinclair ed. *Looking Up.* Collins, 1987, p.121 .

[97] Hansberry, Lorraine. *A raisin in the sun.* (1959)

[98] Herodotus. *On Libya*,from the *Histories*.

[99] *[Letter of Capt. E. Hopkins. Providence, 10 Sep 1764]*

[100] Hornby, A.S. et al. *Oxford Advanced Learner's Dictionary of Current English*. Oxford University Press (1974)

[101] Ibsen, Henrik, tr. William and Charles Archer. *Peer Gynt* (1875)

[102] Ibsen, Henrik, tr. R. Farquharson Sharp and Eleanor Marx-Aveling. *A Doll's House* In *A Doll's House; and Two Other Plays by Henrik Ibsen*, Everyman's library: the drama 494, London: J. M. Dent & Sons, 1910.

[103] Idle, Eric, Michael Palin, Graham Chapman, John Cleese, Terry Gilliam. *The Complete Monty Python's Flying Circus: All the Words*, Pantheon Books, (1989), 2 , p.230.

[104] *ISO 690: 1987 Information and documentation, Bibliographic references, Content, form and structure* clause 4.1 , p.2.

[105] Jarry, Alfred, tr. Simon Watson Taylor and Cyril Connolly. *The Ubu plays*, London: Methuen, 1968.

[106] Jerome, Jerome K. *Three men in a boat* (1889), chapter 6.

[107] Jonson, Ben. *Volpone*, J. B. Bamborough ed. Macmillan, 1963, p.14.

[108] Jonson, Ben. *The Alchemist* , Douglas Brown ed. London: Benn, 1966, I.1, p.9.

[109] Kersey, John. *Dictionarium Anglo-Brittannicum: Or, a General English Dictionary*. London: J. Wilde, 1715. 2nd ed.

[110] Kipling, Rudyard. `The mother hive'. In *Actions and Reactions*, London: Macmillan, (1909)

[111] Kipling, Rudyard. *Stalky & Co.*, London: Macmillan (1899)

[112] Kipling, Rudyard. *Kim*, Macmillan, (1901), p. 9.

[113] Kyd, Thomas. *Spanish Tragedy* (1592).

[114] La Fontaine, Jean de. `L'Astrologue qui se laisse tomber dans un puits' In *Fables Choisies*, Classiques Larousse, Paris: Librairie Larousse, 1, 1940.

[115] Laclos, Pierre Choderlos de. *Les Liaisons dangereuses* (1972), 1963, p.13.

[116] Ladurie, Emmanuel Le Roy. *Montaillou*, Middlesex: Penguin Books, 1980, p.3.

[117] Langendoen, D. Terence and Paul M. Postal. *Vastness of Natural Languages*, Oxford: Basil Blackwell, 1984, p.24 , note 12.

[118] Langland, William. *The Vision of Piers Plowman* In A.V.C. Schmidt ed. *Langland: Vision of Piers Plowman: "B" Text*, opening.

[119] Lawrence, David Herbert. Autograph manuscript of *Eloi, Eloi, lama sabachthani*, Pierpont Morgan MA 1892; in Klinkenborg and Cahoon (1981) p.129.

[120] Layamon. *Brut*, fol 65v of Bodleian MS. Rawlinson Poetry 32; in Parkes (1969) 12(ii).

[121] Leech, Geoffrey and Mick Short. *Style in Fiction*, London: Longman, 1981, p.272.

[122] Leech, G., G. Myers, J. Thomas eds. *Spoken English on Computer: Transcription, Markup and Applications.* Harlow: Longman, 1995.

[123] Lessing, Doris. *Martha Quest*. 1952, pp. 52-53

[124] LeTourneau, Mark S. *English Grammar*, 2001, New York: Harcourtp. 89.

[125] Lewis, Leopold Davis. *The Bells*, (1871), translated from Erckmann-Chatrian, *Le Juif Polonais*.

[126] Lewis, Wyndham. *Tarr* (1928), Jupiter, 1968, p.17

[127] Lillo, George. *The London Merchant* (1731), epilogue.

[128] Lincoln, Abraham. `A. Lincoln to Richard Yates and William Butler', 10 Apr 1862. In Fehrenbacher, Don E. ed. *Lincoln: Speeches and Writings,* 2 (1859-1865). Library of America46 , 1989, p.315.

[129] Lincoln, Abraham. `Second Inaugural Address', 4 March 1865. In H. S. Commager, ed., *Documents of American History*, 5th ed Crofts American history series. New York: Appleton-Century-Crofts, 1949, p.442.

[130] *Longman Dictionary of Contemporary English*. Harlow, Essex: Longman (1978)

[131] Lowe, David. *Lost Chicago*, Boston: Houghton Mifflin, (1978), p.30.

[132] Luther, Martin [tr]. *Die gantze Heilige Schrifft Deudsch, Wittenberg 1545. Letzte zu Luthers Lebzeiten erchienene Ausgabe, hsg. Hans Volz unter Mitarbeit von Heinz Blanke. Textredaktion Friedrich Kur*, München: Rogner & Bernhard, 1972.

[133] MacNeice, Louis. `The Sunlight on the Garden' In E.R. Dodds, *The collected poems of Louis MacNeice*, London: Faber, 1966.

[134] Examples from MacWhinney, Brian, 88, 87, cited by Johansson, S. `The approach of the Text Encoding Initiative to the encoding of spoken discourse' In Leech, G., G. Myers, J. Thomas eds. *Spoken English on Computer: Transcription, Markup and Applications*. Harlow: Longman, 1995.

[135] Madan, Falconer, et al, *A summary catalogue of western manuscripts in the Bodleian Library at Oxford which have not hitherto been catalogued ...* Oxford, 1895-1953. 5: 515. (Cited in Driscoll, M.J. `P5-MS: A general purpose tagset for manuscript description' in *Digital Medievalist*: 2.1, (2006))

[136] Any issue of the *Malawi Daily Times.*

[137] `The Manere of Good Lyuynge': fol. 126v of Bodleian MS Laud Misc 517; in Parkes (1969), p.8.

[138] *Marbury v. Madison*, 1 Cranch, 137 (1803), rpt. In H. S. Commager, ed., *Documents of American History*, 5th edCrofts American history series. New York: Appleton-Century-Crofts, 1949, p.192.

[139] Marvell, Andrew. *An Horatian Ode*, Bod. MS Eng. Poet d.49.

[140] Melville, Herman, *Moby Dick*. (1851).

[141] Miller, Henry. `Death of a Salesman' in Atkinson, Brooks, *New Voices in the American Theatre*, New York: Modern Library, 1955, p.113.

[142] Milne, A. A. *The House at Pooh Corner*. London: Methuen & Co., 1928, p.83.

[143] Milton, John. *Paradise Lost: A poem in X books*. (1667), I, 1-10

[144] Moore, George. Autograph manuscript of *Memoirs of my dead life*, Pierpont Morgan MA 3421; in Klinkenborg and Cahoon (1981)

[145] Moore, Thomas. Autograph manuscript of the second version of *Lalla Rookh*, Pierpont Morgan MA 310; in Klinkenborg and Cahoon (1981) 23.

[146] Moreland, Floyd L. and Rita M. Fleischer. *Latin: An Intensive Course,* (1977) p.53.

[147] *The New Penguin English Dictionary*. London: Penguin Books (1986)

[148] *Njal's saga*. tr. Magnus Magnusson and Hermann Palsson. Penguin. (1960), chapter 12, p.60.

[149] O'Casey, Sean. *Time to go*. (1951)

[150] Orwell, George. *Nineteen-Eighty-Four*. London: Gollancz (1949)

[151] Owen, Wilfred. *Dulce et decorum est*, from autograph manuscript in the English Faculty Library, Oxford University.

[152] Payne, J. `Report on the compatibility of J P French's spoken corpus transcription conventions with the TEI guidelines for transcription of spoken texts', *Working Paper*, Dec 1992, NERC WP8/WP4 122.

[153] Peacock, Thomas Love. *Gryll Grange* (1861), chapter 1.

[154] `Partial family tree for Bertrand Russell' based on an example in Pereira, Fernando C.N. and Stuart M. Shieber, *Prolog and Natural Language Analysis*, Stanford: Center for the Study of Language and Information. (1987), p.22.

[155] *Petit Larousse en Couleurs*. Paris: Larousse, (1990)

[156] Pinsky, Robert. `Essays on Psychiatrists' in *Sadness and Happiness* (1975)

[157] Plautus, Titus Macchius. *Menaechmi*.

[158] Pope, Alexander. `The Rape of the Lock' (1714) III.7.

[159] Pope, Alexander. *An Essay on Criticism* (1711).

[160] Pope, Alexander. *Dunciad Variorum* (1729), III.284.

[161] From letter 'JK' found in *Poulson's Daily Advertiser,* 8 Oct 1835.

[162] Queneau, Raymond. *Exercices de style*. Paris: Gallimard, (1947), p.192.

[163] Reference from the bibliography in Reps, Thomas William and Teitelbaum, Tim eds. *The Synthesizer Generator: A system for constructing language-based editors*, New York and Berlin: Springer-Verlag, 1989, p.304.

[164] Richardson, Samuel. *Clarissa; or the History of a Young Lady*. (1748), 2 Letter XIV.

[165] Robert, Paul. *Le Petit Robert*. Paris: Dictionnaires Le Robert (1967)

[166] Rowling, J. K.`The Sorting Hat'. In *Harry Potter and the Sorcerer's Stone*, (1999), chapter 7 , p.121.

[167] *The Saga of the Volsungs: the Norse epic of Sigurd the Dragon Slayer*. trans. Jesse L. Byock. University of California Press (1990).

[168] Sapir, Edward. *Language: an introduction to the study of speech*, New York: Harcourt, Brace and World, 1921, p.79.

[169] Shakespeare, William. *Henry V* In *Mr. William Shakespeares Comedies, Histories, & Tragedies*, London: Jaggard and Blount, 1623.

[170] Shakespeare, William. *Henry V* In *The Works of Shakespeare in seven volumes*, ed. Lewis Theobald, London: Bettesworth, Hitch, Tonson et al, (1733).

[171] Shakespeare, William. *Antony and Cleopatra*, IV.4, 14-21

[172] Shakespeare, William. *Merchant of Venice*, I.ii, speech 5 (Portia).

[173] Shakespeare, William. *Macbeth*, Act V, Scene 1.

[174] Shakespeare, William. *Antony and Cleopatra* (1623), V.ii.

[175] Shakespeare, William. `Hamlet' In Stanley Wells and Gary Taylor eds. *The Complete Works*, Oxford: Clarendon Press, 1986, I.i.

[176] Shakespeare, William. *Hamlet*, London: Valentine Simmes. (1603), I.i.

[177] Shakespeare, William. `The Tempest ' In *Mr. William Shakespeares Comedies, Histories, & Tragedies*, London: Jaggard and Blount, (1623).

[178] Shakespeare, William, *The Sonnets* (1609), 130.

[179] Shaw, George Bernard. *Heartbreak House: a fantasia in the Russian manner on English themes.* (1916)

[180] Shaw, George Bernard. *Pygmalion*, 1913.

[181] Shields, David. *Dead Languages*, HarperCollins Canada/Perennial Rack, rpt. 1990, p.10.

[182] Smith, Adam. *An Inquiry into the Nature and Causes of the Wealth of Nations*, London. (1776), index to vol. 1.

[183] Smith, Sydney. *Autograph letter*. In Pierpont Morgan library; Klinkenborg and Cahoon (1981) 11.

[184] Southey, Robert. Autograph manuscript of *The Life of Cowper*. In Pierpont Morgan MA 412 (Klinkenborg and Cahoon (1981) 15).

[185] Soyinka, Akinwande Oluwole Wole. *Madmen and Specialists*, London: Methuen (1971)

[186] Spenser, Edmund. *The Faerie Queene: Disposed into twelue bookes, Fashioning XII. Morall vertues.* (1596)

[187] Sterne, Laurence. *The Life and Opinions of Tristram Shandy, Gentleman*. (1760)

[188] Swift, Jonathan. *Travels into Several Remote Nations of the World, in Four Parts. By Lemuel Gulliver...* . (1735)

[189] Swinburne, Algernon Charles. *Poems and Ballads (First Series)*. London: Chatto & Windus. (1904)

[190] Swinnerton, Frank Arthur. *The Georgian Literary Scene 1910-1935*, 1938, London: J. M. Dent, p.195.

[191] Sutherland, L.S. and L.G. Mitchell eds. *The Eighteenth century*, The History of the University of Oxford V, p.178.

[192] *The Guardian,* 26 Oct 1992, p.2

[193] *The Guardian,* 21 Dec 1992, p.2.

[194] *The Independent,* 26 Oct 1775, headline.

[195] Thurber, James. *The 13 Clocks* (1950).

[196] Townsend, Sue. *The growing pains of Adrian Mole*. (1984), p.43

[197] Trollope, Anthony. *An Autobiography* (1883).

[198] Trollope, Anthony, *North America*. (1862)

[199] Tufte, Edward R., *Envisioning Information*, Cheshire: Graphics Press. (1990)

[200] *United States Code* Title 17, Section 107, found at `http://www.copyright.gov/title17/92chap1.html#107`

[201] United States District Court for the Middle District of Pennsylvania. *Kitzmiller v. Dover Area School District et al.*: 2005. 04cv2688, p. 33. Available from `http://www.pamd.uscourts.gov/kitzmiller/kitzmiller_342.pdf` and transcribed at `http://en.wikisource.org/wiki/Kitzmiller_v._Dover_Area_School_District_et_al.`.

[202] Vergil (Publius Vergilius Naso). *Aeneid*, I.1

[203] *Vóluspá* recto of folio 5 of the unique manuscript of the Elder Edda. Codex Regius, ed. L. F. A. Wimmer and F. Jónsson (Copenhagen 1891).

[204] Wanton, Joseph. *Unpublished letter to Nicholas Brown and Co*, 1761Brown University Steering Committee on Slavery and Justice: Repository of Historical Documents. (`http://dl.lib.brown.edu/slaveryandjustice/`).

[205] Wanklyn, M.D.G. et al. *Gloucester Port Books, 1575-1765*. Available from `http://www.ahds.ac.uk/catalogue/collection.htm?uri=hist-3218-1`

[206] Warriner, John E. *English Composition and Grammar*. (1988), p.280.

[207] *Webster's Seventh Collegiate Dictionary*. Springfield, Mass. G. & C. Merriam Co. (1975)

[208] Williams, Nigel. *The Wimbledon Poisoner* (1990), p.204.

[209] Woolf, Virginia, *Mrs Dalloway* (1925), p.65.

[210] Wordsworth, William. `Scorn not the sonnet' in *Poetical Works*. (1827)

[211] Wordsworth, William. *The Prelude*. (1850)

[212] Wycherley, William. *The Country Wife* (1675).

[213] `Zuigan calls himself "Master"'. Mumon Ekai. (In *The Gateless Gate*, Case 12.)

Works cited elsewhere in the text of the Guidelines

[Amsler and Tompa (1988)] Robert A. Amsler, Frank W. Tompa. 'An SGML-Based Standard for English Monolingual Dictionaries'. *Information in Text*, *Fourth Annual Conference of the U[niversity of] W[aterloo] Centre for the New Oxford English Dictionary*, (Fourth Annual Conference of the U[niversity of] W[aterloo] Centre for the New Oxford English Dictionary, October 26-28, 1988, Waterloo, Canada) October 1988. Waterloo, Canada. pp. 61-79.

[Berglund (ed.) (2006)] Anders Berglund (ed.) *Extensible Stylesheet Language (XSL) Version 1.1*, 5 December 2006. W3C. <`http://www.w3.org/TR/xsl11`>.

[Bray et al. (eds.) (2006)] Tim Bray, Jean Paoli, C. M. Sperberg-McQueen, Eve Maler, François Yergau (eds.) *Extensible Markup Language (XML) Version 1.0 (Fourth edition)*, 16 August 2006. W3C. <`http://www.w3.org/TR/REC-xml/`>.

[Bray et al. (eds.) (2006)] Tim Bray, Dave Hollander, Andrew Laymon, Richard Tobin (eds.) *Namespaces in XML 1.0 (second edition)*, 16 August 2006. W3C. <`http://www.w3.org/TR/xml-names/`>.

[Klinkenborg and Cahoon (1981)] *British Literary Manuscripts. Series 2: from 1800 to 1914*, Verlyn Klinkenborg, Herbert Cahoon. 1981. New York: Pierpont Morgan Library.

[Burnard (1988)] Lou Burnard. 'Report of Workshop on Text Encoding Guidelines'. *Literary & Linguistic Computing* 1988. 3.

[Burnard and Sperberg-McQueen (1995)] Lou Burnard, C. Michael Sperberg-McQueen. 'The Design of the TEI Encoding Scheme'. *Computers and the Humanities* 1995. 29 (1) p. 17–39. <`http://dx.doi.org/10.1007/BF01830314`>.(Reprinted in Ide and Veronis (eds.) (1995), pp. 17-40)

[Burnard and Rahtz (2004)] Lou Burnard, Sebastian Rahtz. *RelaxNG with Son of ODD*, *Proceedings of Extreme Markup Languages 2004*, 2004. <http://www.mulberrytech.com/Extreme/Proceedings/html/2004/Burnard01/EML2004Burnard01.pdf>.

[Burrows (1987)] John Burrows. *Computation into Criticism: A Study of Jane Austen's Novel and an Experiment in Method*, 1987. Oxford: Clarendon Press.

[Calzolari et al. (1990)] N. Calzolari, C. Peters, A. Roventini. *Computational Model of the Dictionary Entry: Preliminary Report*, *Acquilex: Esprit Basic Research Action No. 3030, Six-Month Deliverable*, April 1990. Pisa.

[Carlisle et al. (eds.) (2003)] David Carlisle, Patrick Ion, Robert Miner, Nico Poppelier (eds.) *Mathematical Markup Language (MathML) Version 2.0 (Second edition)*, 21 October 2003. W3C. <http://www.w3.org/TR/MathML2/>.

[Carpenter (1992)] Bob Carpenter. *The logic of typed feature structures*, 1992. Cambridge: Cambridge University Press. Cambridge Tracts in Theoretical Computer Science 32.

[Chartrand and Lesniak (1986)] Gary Chartrand, Linda Lesniak. *Graphs and Digraphs*, 1986. Menlo Park, CA: Wadsworth.

[Chatti et al. (2007)] Noureddine Chatti, Suha Kaouk, Sylvie Calabretto, Jean Marie Pinon. 'MultiX: an XML based formalism to encode multistructured documents'. *Proceedings of Extreme Markup Languages 2007*, 2007. <http://www.idealliance.org/papers/extreme/proceedings/html/2007/Chatti01/EML2007Chatti01.html>.

[Clark (ed.) (1999)] James Clark (ed.) *XSL Transformations (XSLT) Version 1.0*, 16 November 1999. W3C. <http://www.w3.org/TR/xslt/>.

[Clark and DeRose (eds.) (1999)] James Clark, Steve DeRose (eds.) *XML Path Language (XPath) Version 1.0*, 16 November 1999. W3C. <http://www.w3.org/TR/xpath/>.

[DANLEX Group (1987)] The DANLEX Group. 'Descriptive tools for electronic processing of dictionary data'. *Lexicographica, Series Maior* 1987. Tübingen: Niemeyer.

[Davis et al. (2006)] Mark Davis, Ken Whistler, Asmus Freytag. *Unicode Character Database*, 2006. Unicode Consortium. <http://www.unicode.org/Public/UNIDATA/UCD.html>.

[Dekhtyar and Iacob (2005)] Alex Dekhtyar, Ionut E. Iacob. *A framework for management of concurrent XML markup*, 2005. <http://www.eppt.org/~emil/publications/dke04-concurrent.pdf>.

[DeRose (2004)] Steven DeRose. 'Markup overlap: a review and a horse'. *Proceedings of Extreme Markup Languages 2004*, 2004. <http://www.mulberrytech.com/Extreme/Proceedings/html/2004/DeRose01/EML2004DeRose01.html>.

[Durusau and O'Donnell (2002)] Patrick Durusau, Matthew Brook O'Donnell. 'Coming down from the trees: next step in the evolution of markup?'. *Proceedings of Extreme Markup Languages 2002*, 2002.

[Edwards and Lampert (eds.) (1993)] J. A. Edwards, M. D. Lampert (eds.) *Talking Language: Transcription and Coding of Spoken Discourse*, 1993. Hillsdale, N.J.: Lawrence Erlbaum Associates.

[Fought and Van Ess-Dykema] John Fought, Carol Van Ess-Dykema. *Toward an SGML Document Type Definition for Bilingual Dictionaries*, *TEI working paper TEI AIW20*, available from the TEI..

[Freytag (2006)] Asmus Freytag. *The Unicode Character Property Model*, Unicode Technical Report #23 2006. <http://www.unicode.org/reports/tr23/>.

[Gale and Church (1993)] William A. Gale, Kenneth W. Church. 'Program for aligning sentences in bilingual corpora'. *Computational Linguistics* 1993. 19 pp. 75-102.

[Garside et al. (1991)] R. G. Garside, G. N. Leech, G. R. Sampson. *The Computational Analysis of English: a Corpus-Based Approach*, 1991. Oxford: Oxford University Press.

[Gorman and Winkler (eds.) (1978)] Michael Gorman, Paul W. Winkler (eds.) *Anglo-American Cataloguing Rules*, Second Edition. 1978. Chicago, London, Ottawa: American Library Association. Library Association. Canadian Library Association.

[Grosso et al. (eds.) (2003)] Paul Grosso, Eve Maler, Jonathan Marsh, Norman Walsh (eds.) *XPointer Framework*, 25 March 2003. W3C. <http://www.w3.org/TR/xptr-framework/>.

[Grosso et al. (eds.) (2003)] Paul Grosso, Eve Maler, Jonathan Marsh, Norman Walsh (eds.) *XPointer element() Scheme*, 25 March 2003. W3C. <http://www.w3.org/TR/xptr-element/>.

[Hilbert et al. (2005)] Mirco Hilbert, Oliver Schonefeld, Andreas Witt. 'Making CONCUR work'. *Proceedings of Extreme Markup Languages 2005*, 2005. <http://www.mulberrytech.com/Extreme/Proceedings/html/2005/Witt01/EML2005Witt01.xml>.

[Huitfeldt and Sperberg-McQueen (2001)] Claus Huitfeldt, C. Michael Sperberg-McQueen. *TexMECS: An experimental markup meta-language for complex documents*, 2001. <http://decentius.aksis.uib.no/mlcd/2003/Papers/texmecs.html>.

[Ide et al. (1992)] Nancy Ide, Jean Veronis, Susan Warwick-Amstrong, Nicoletta Calzolari. 'Principles for Encoding machine readable dictionaries'. *Proceedings of the Fifth EURALEX International Congress, EURALEX'92*, (Fifth EURALEX International Congress, EURALEX'92, University of Tampere, Finland) 1992.

[Ide et al. (1993)] Nancy Ide, Jacques Le Maitre, Jean Veronis. 'Outline of a Model for Lexical Databases'. *Information Processing and Management* 1993. 29 (2) pp. 159-186.

[Ide and Veronis (1995)] Nancy Ide, Jean Veronis. 'Encoding Print Dictionaries'. *Computers and the Humanities* 1995. 29 pp. 167-195.

[Ide et al. (2000)] N. Ide, A. Kilgarriff, L. Romary. 'A Formal Model of Dictionary Structure and Content'. *Proceedings of Euralex 2000*, (Euralex 2000) 2000. Stuttgart. pp. 113-126.

[Jackendoff (1977)] R. Jackendoff. 'X-Bar Syntax: A study of phrase structure'. *Linguistic Inquiry Monograph* 1977. 2.

[Jagadish et al. (2004)] H. V. Jagadish, Laks V. S. Lakshmanan, Monica Scannapieco, Divesh Srivastava, Nuwee Wiwatwattana. *Colorful XML: one hierarchy isn't enough*, 2004. <http://www.research.att.com/~divesh/papers/jlssw2004-mct.pdf>.

[Johansson et al. (1991)] Stig Johansson, Lou Burnard, Jane Edwards, And Rosta. *Working Paper on Spoken Texts*, *TEI document TEI AI2 W1*, 1991.

[Johansson (1994)] Stig Johansson. 'Encoding a Corpus in Machine-Readable Form'. Sue Atkins, Antonio Zampolli (eds.) *Computational Approaches to the Lexicon: An Overview*, 1994. Oxford: Oxford University Press.

[Knuth (1992)] Donald E. Knuth. *Literate Programming*, CSLI Lecture Notes 27 1992. Stanford, California: Center for the Study of Language and Information. 0-937073-80-6

[Kytö and Rissanen (1988)] M. Kytö, M. Rissanen. 'The Helsinki Corpus of English Texts'. M. Kytö, O. Ihalainen, M. Rissanen (eds.) *Corpus Linguistics: hard and soft*, 1988. Amsterdam: Rodopi.

[Langendoen and Simons (1995)] D. Terence Langendoen, Gary F. Simons. 'A rationale for the TEI recommendations for feature-structure markup'. *Computers and the Humanities* 1995. 29 pp. 167-195.

[Leech and Garside (1991)] G. N. Leech, R. G. Garside. 'Running a Grammar Factory'. S. Johansson, A.-B. Stenstrøm (eds.) *English Computer Corpora: Selected Papers and Research Guide*, 1991. Berlin, New York: de Gruyter. Mouton. pp. pp. 15-32..

[Lie and Bos (eds.) (1999)] Håkon Wium Lie, Bert Bos (eds.) *Cascading Style Sheets, Level 1*, 11 January 1999. W3C. <http://www.w3.org/TR/REC-CSS1/>.

[Loman and Jørgensen (1971)] Bengt Loman, Nils Jørgensen. *Manual for analys och beskrivning av makrosyntagmer*, 1971. Lund: Studentlitteratur.

[MacWhinney (1988)] Brian MacWhinney. *CHAT Manual*, 1988. Pittsburgh: Dept of Psychology, Carnegie-Mellon University. pp. 87ff.

[Marsh (ed.) (2001)] Jonathan Marsh (ed.) *XML Base*, 27 June 2001. W3C. <http://www.w3.org/TR/xmlbase/>.

[Marshall (1983)] I. Marshall. 'Choice of Grammatical Word Class without Global Syntactic Analysis: Tagging Words in the LOB Corpus'. *Computers and the Humanities* 1983. 17 pp. 139-50.

[Mattheier et al. (eds.) (1988)] Klaus Mattheier, Ulrich Ammon, Peter Trudgill (eds.) *Sociolinguistics*, *Soziolinguistik*, *An international handbook of the science of language and society*, *Ein internationales Handbuch zur Wissenschaft von Sprache und Gesellschaft*, 1988. Berlin, New York: De Gruyter. I pp. 271 and 274.

[Parkes (1969)] M. B. Parkes. *English Cursive Book Hands 1250–1500*, 1969. Oxford: Clarendon Press.

[Pereira (1987)] Fernando C. N. Pereira. *Grammars and logics of partial information*, 1987. Menlo Park, CA: SRI International. SRI International Technical Note 420.

[Petty (1977)] A. G. Petty. *English literary hands from Chaucer to Dryden*, 1977. London: Edward Arnold. pp. 22–25.

[Phillips and Davis (eds.) (2006)] Addison Phillips, Mark Davis (eds.) *Tags for Identifying Languages*, 2006. IETF. RFC 4646

[Phillips and Davis (eds.) (2006)] Addison Phillips, Mark Davis (eds.) *Matching of Language Tags*, 2006. IETF. RFC 4647

[Ragget et al. (eds.) (1999)] Dave Ragget, Arnaud Le Hors, Ian Jacobs (eds.) *HTML 4.01 Specification*, 24 December 1999. W3C. <http://www.w3.org/TR/html401/>.

[Renear et al. (1996)] A. Renear, E. Mylonas, D. Durand. 'Refining our notion of what text really is: the problem of overlapping hierarchies'. Nancy Ide, Susan Hockey (eds.) *Research in Humanities Computing*, 1996. Oxford University Press.

[Shieber (1986)] Stuart Shieber. *An Introduction to Unification-based Approaches to Grammar*, 1986. Palo Alto, CA: Center for the Study of Language and Information. CSLI Lecture Notes 4

[Tennison and Piez (2002)] Jeni Tennison, Wendell Piez. 'The layered markup and annotation language'. *Proceedings of Extreme Markup Languages Conference*, 2002.

[Unicode Consortium (2006)] *The Unicode Standard, Version 5.0*, Unicode Consortium. 2006. Addison-Wesley Professional. <http://www.unicode.org/>.

[Tutin and Veronis (1998)] Agnès Tutin, Jean Veronis. 'Electronic dictionary encoding: customizing the TEI Guidelines'. *Proceedings of the Eighth Euralex International Congress*, (Eighth Euralex International Congress) 1998.

[van der Vlist (2004)] Eric van der Vlist. *RELAX NG*, 2004. O'Reilly.

[Witt (2002)] Andreas Witt. *Multiple Informationsstrukturierung mit Auszeichnungssprachen. XML-basierte Methoden und deren Nutzen für die Sprachtechnologie*, 2002. (Ph D thesis, Bielefeld University) (See also http://xml.coverpages.org/Witt-allc2002.html)

[XHTML™ 1.0 The Extensible HyperText Markup Language (Second Edition) (2000)] *XHTML™ 1.0 The Extensible HyperText Markup Language (Second Edition)*, 26 January 2000. W3C. <http://www.w3.org/TR/xhtml/>.

Reading list

The following lists of readings in markup theory and the TEI derive from work originally prepared by Susan Schreibman and Kevin Hawkins for the TEI Education Special Interest Group, recoded in TEI P5 by Sabine Krott and Eva Radermacher. They should be regarded only as a snapshot of work in progress, to which further contributions and corrections are welcomed (see further http://www.tei-c.org/Activities/SIG/Education/tei_bibliography.xml).

Theory of Markup and XML

[Barnard et al. (1995)] David T. Barnard, Lou Burnard, Jean-Pierre Gaspart, Lynne A. Price, C. Michael Sperberg-McQueen, Giovanni Battista Varile. 'Hierarchical Encoding of Text: Technical Problems and SGML Solutions'. *Computers and the Humanities* 1995. 29 (3) p. 211–231. <http://dx.doi.org/10.1007/BF01830617>. <http://www.tei-c.org/Vault/ML/mlw18.ps>.

[Barnard et al. (1996)] David T. Barnard, Lou Burnard, C. Michael Sperberg-McQueen. 'Lessons learned from using SGML in the Text Encoding Initiative'. *Computer Standards & Interfaces* 1996. 18 (1) p. 3–10. <http://dx.doi.org/10.1016/0920-5489(95)00035-6>.

[Burnard (1991)] Lou Burnard. 'What is SGML and how does it help?'. Daniel Greenstein (ed.) *Modelling Historical Data: Towards a Standard for Encoding and Exchanging Machine-readable Texts*, 1991. St Katherinen: Max-Planck-Institut für Geschichte In Kommission bei Scripta Mercaturae Verlag. p. 81–91. Halbgraue Reihe zur Historischen Fachinformatik Herausg. vonManfred Thallerserie A 11. <http://www.tei-c.org/Vault/ED/EDW25/>.(Revised version published as Burnard (1995))

[Burnard (1995)] Lou Burnard. 'SGML on the Web: Too Little Too Soon or Too Much Too Late?'. *Computers & Texts* 1995. 15 p. 12–15. <http://users.ox.ac.uk/~lou/Belux/>.

[Burnard (1995)] Lou Burnard. 'What is SGML and How Does It Help?'. *Computers and the Humanities* 1995. 29 (1) p. 41–50. <http://dx.doi.org/10.1007/BF01830315>. <http://xml.coverpages.org/burnardw25-index.html>.(Reprinted in Ide and Veronis (eds.) (1995), pp. 41-50)

[Burnard (1999)] Lou Burnard. *Is Humanities Computing an Academic Discipline? or, Why Humanities Computing Matters*, 1999. <http://www.iath.virginia.edu/hcs/burnard.html>.(Presented at an interdisciplinary seminar at the Institute for Advanced Technology in the Humanities, University of Virginia, November 1999.) <http://www.iath.virginia.edu/hcs/>.

[Burnard (1999)] Lou Burnard. 'Using SGML for Linguistic Analysis: The Case of the BNC'. *Markup Languages Theory and Practice* 1999. Cambridge, Massachusettes: MIT Press. 2 p. 31–51. <http://users.ox.ac.uk/~lou/papers/sgml96.sgm>.(Also published in Moser et al. (eds.) (2001), pp. 53–72)

[Burnard et al. (1999)] Lou Burnard, Elizabeth Lalou, Peter Robinson. 'Vers un Standard Européen de Description des Manuscrits: Le Projet Master'. *Documents Numeriques Les Documents Anciens* 1999. Paris: Hermes Science Publications. 3 (1–2) pp. 151-169.

[Burnard (1999)] Lou Burnard. *XML: The Dream and the Reality*, 1999. <http://users.ox.ac.uk/~lou/papers/euro99.xml>.(Closing plenary address at the XML Europe Conference, Granada, May 1999)

[Burnard et al. (2000)] Lou Burnard, Claudia Claridge, Josef Schmied, Rainer Siemund. 'Encoding the Lampeter Corpus'. *DRH98: Selected Papers from Digital Resources for the Humanities*, 2000. London: Office for Humanities Communication. <http://users.ox.ac.uk/~lou/papers/glasgie.xml>.

[Burnard (2000)] Lou Burnard. *From Two Cultures to Digital Culture: The Rise of the Digital Demotic*, 2000. (Presented at CLIP, Alicante) <http://users.ox.ac.uk/~lou/wip/twocults.html>.(Published in Italian as Burnard (2001))

[Burnard (2001)] Lou Burnard. 'Dalle «Due Culture» Alla Cultura Digitale: La Nascita del Demotico Digitale'. Translated byFederico Pellizi*Il Verri Nella Rete* 2001. Milano: Monogramma. 16 p. 9–22.

[Burnard (2001)] Lou Burnard. 'On the Hermeneutic Implications of Text Encoding'. Domenico Fiormonte, Jonathan Usher (eds.) *New Media and the Humanities: Research and Applications*, 2001. Oxford: Humanities Computing Unit. p. 31–38. <http://users.ox.ac.uk/~lou/wip/herman.htm>.

[Burnard (2005)] Lou Burnard. 'Encoding Standards for the Electronic Edition'. Matija Ogrin (ed.) *Znanstvene Izdaje in Elektronski Medij*, *Scholarly Editions and the Digital Medium*, 2005. Ljubljana: Studia Litteraria ZRC ZAZU. p. 12–67. <http://nl.ijs.si/e-zrc/bib/eziss-Burnard.pdf>.

[Burnard (2005)] Lou Burnard. 'Metadata for corpus work'. Martin Wynne (ed.) *Developing Linguistic Corpora: A Guide to Good Practice*, 2005. Oxford: Oxbow Books. p. 30–46. <http://users.ox.ac.uk/~lou/wip/metadata.html>.

[Burnard et al. (eds.) (2006)] Lou Burnard, Katherine O'Brien O'Keefe, John Unsworth (eds.) *Electronic Textual Editing*, 2006. New York: Modern Languages Association. <http://www.tei-c.org/Activities/ETE/>.

[Buzzetti (2002)] Dino Buzzetti. 'Digital Representation and the Text Model'. *New Literary History* 2002. 33 (1) p. 61–88. <http://muse.jhu.edu/journals/new_literary_history/v033/33.1buzzetti.html>.

[Caton (2001)] Paul Caton. 'Markup's Current Imbalance'. *Markup Languages: Theory and Practice* 2001. 3 (1) p. 1–13. (This paper was proceeded by reports at the Joint Annual Conference of the Association for Computers and the Humanities and the Association for Literary and Linguistic Computing in 1999 (Charlottesville, Virginia) and Extreme Markup Languages 2000 (Montreal, Canada))

[Chen and Yu (2003)] Ruey-Shun Chen, Shien-Chiang Yu. 'Developing an XML Framework for Metadata System'. *Proceedings of the 1st International Symposium on Information and Communication Technologies*, 2003. Dublin. p. 267–272. ACM International Conference Proceeding Series 49. <http://portal.acm.org/citation.cfm?id=963653>.(This paper was presented in a session entitled "Electronic Document Technology.")

[Coombs (1986)] James H. Coombs. *Information Management System for Scholars*, 1986. Providence: Brown Computer Center. Technical Memorandum TM 69–2

[Coombs et al. (1987)] James H. Coombs, Allen Renear, Steven J. DeRose. 'Markup Systems and The Future of Scholarly Text Processing'. *Communications of the ACM* 1987. 30 (11) p. 933–947. <http://doi.acm.org/10.1145/32206.32209>. <http://xml.coverpages.org/coombs-hallgren.html>. <http://xml.coverpages.org/coombs.html>.(Reprinted with new commentary in Landow and Delany (eds.) (1993), pp 85–118)

[Cover (2005)] Robin Cover. *Markup Languages and (Non-) Hierarchies*, 2005. (Technology report from the Cover Pages) <http://xml.coverpages.org/hierarchies.html>.

[DeRose et al. (1990)] Steven J. DeRose, David G. Durand, Elli Mylonas, Allen H. Renear. 'What is Text, Really?'. *Journal of Computing in Higher Education* 1990. 1 (2) p. 3–26. (Republished (DeRose et al. (1997)) as a "classic reprint" with invited commentary and authors' replies in the ACM/SIGDOC)

[DeRose (1995)] Steven J. DeRose. *Structured Information: Navigation, Access, and Control*, 1995. (Paper presented at the Berkeley Finding Aid Conference, April 4–6, 1995) <http://sunsite.berkeley.edu/FindingAids/EAD/derose.html>.

[DeRose et al. (1997)] Steven J. DeRose, David G. Durand, Elli Mylonas, Allen H. Renear. 'What is Text, Really?'. *Journal of Computer Documentation* 1997. 21 (3) p. 1–24. <http://doi.acm.org/10.1145/264842.264843>.

[Goldfarb (1981)] Charles F. Goldfarb. 'A Generalized Approach to Document Markup'.. *Proceedings of the ACM SIGPLAN SIGOA Symposium on Text Manipulation*, 1981. New York: ACM. 68–73. (Adapted as "Annex A. Introduction to Generalized Markup" in ISO 8879) <http://www.nyct.net/~aray/notes/igm.html>.

[Graham (1999)] Tony Graham. 'Unicode: What Is It and How Do I Use It?'. *Markup Languages: Theory & Practice* 1999. 1 (4) p. 75.

[Hockey (1996)] Susan Hockey. 'Creating and Using Electronic Editions'. Richard J. Finneran (ed.) *The Literary Text in the Digital Age*, 1996. Ann Arbor, MI: University of Michigan Press. p. 1–22.

[Hockey et al. (1999)] Susan Hockey, Allen Renear, Jerome J. McGann. *What is Text? A Debate on the Philosophical and Epistemological Nature of Text in the Light of Humanities Computing Research*, 1999. (Panel presented at ACH/ALLC 1999) <http://www.iath.virginia.edu/ach-allc.99/proceedings/hockey-renear2.html>.

[Hockey (2000)] Susan Hockey. *Electronic Texts in the Humanities*, 2000. New York, NY: Oxford University Press.

[Huitfeldt (1994)] Claus Huitfeldt. 'Multi-dimensional Texts in a One-dimensional Medium'. *Computers and the Humanities* 1994. 28 (4/5) p. 235–241. <http://dx.doi.org/doi:10.1007/BF01830270>.

[Huitfeldt (1994)] Claus Huitfeldt. 'Toward a Machine-Readable Version of Wittgenstein's Nachlaß: Some Editorial Problems'. Hans Gerhard Senger (ed.) *Philosophische Editionen. Erwartungen an sie – Wirkungen durch sie*, 1994. Tübingen: Max Niemeyer Verlag. p. 37–43. Beihefte zu editio 6.

[Ide and Veronis (eds.) (1995)] Nancy Ide, Jean Veronis (eds.) *The Text Encoding Initiative: Background and Contexts*, 1995. Dordrecht, Boston: Kluwer Academic Publisher.

[Lamport (1987)] Leslie Lamport. 'Document Production: Visual or Logical?'. *Notices of the American Mathematical Society* 1987. 34 p. 621–624. <http://research.microsoft.com/users/lamport/pubs/pubs.html#document-production>.(Republished as Lamport (1988))

[Lamport (1988)] Leslie Lamport. 'Document Production: Visual or Logical?'. *TUGboat* 1988. 9 (1) pp. 8-10. <http://www.tug.org/TUGboat/Articles/tb09-1/tb20lamport.pdf>.

[Landow and Delany (eds.) (1993)] George P. Landow, Paul Delany (eds.) *The Digital Word: Text-based Computing in the Humanities*, 1993. Cambridge, MA: MIT Press.

[Lavagnino (1996)] John Lavagnino. 'Completeness and Adequacy in Text Encoding'. Richard J. Finneran (ed.) *The Literary Text in the Digital Age*, 1996. Ann Arbor, MI: University of Michigan Press. p. 63–76.

[Lightfoot (1979)] Charles Lightfoot. *Generic Textual Element Identification—A Primer*, 1979. Arlington: Graphic Communications Computer Association.

[Lubell (1999)] Joshua Lubell. 'Structured Markup on the Web: A Tale of Two Sites'. *Markup Languages: Theory & Practice* 1999. 1 (3) p. 7–22. <http://www.mel.nist.gov/msidlibrary/doc/mlang/markuplang.htm>.

[McEnery et al. (1998)] Tony McEnery, Lou Burnard, Andrew Wilson, Paul Baker. *Validation of Linguistic Corpora*, 1998. <http://users.ox.ac.uk/~lou/wip/ELRA/WP3/>.(Report commissioned by ELRA)

[McGann (1997)] Jerome McGann. 'The Rationale of Hypertext'. Kathryn Sutherland (ed.) *Electronic Text: Investigations in Method and Theory*, 1997. New York, NY: Clarendon Press Oxford. p. 19–46.

[McGann (2001)] Jerome McGann. *Radiant Textuality: Literature After the World Wide Web*, 2001. New York, NY: Palgrave Macmillian.

[McGann (2004)] Jerome McGann. 'Marking Texts of Many Dimensions'. Susan Schreibman, Ray Siemens, John Unsworth (eds.) *A Companion to Digital Humanities*, 2004. Oxford: Blackwell. p. 198–217. <http://www.digitalhumanities.org/companion/>.

[Morrison et al. ((no date))] Alan Morrison, Michael Popham, Karen Wikander. *Creating and Documenting Electronic Texts: A Guide to Good Practice*, (no date). <http://ota.ahds.ac.uk/documents/creating/>.

[Moser et al. (eds.) (2001)] Stephan Moser, Peter Stahl, Werner Wegstein, Norbert Richard Wolf (eds.) *Maschinelle Verarbeitung Altdeutscher Texte V (Beiträge zum Fünften Internationalen Symposion, Würzburg, 4–6 März 1997)*, 2001. Tübingen: Niemeyer.

[Pichler (1995)] Alois Pichler. 'Advantages of a Machine-Readable Version of Wittgenstein's Nachlaß'. Kjell S. Johannessen, Tore Nordenstam (eds.) *Culture and Value: Philosophy and the Cultural Sciences. Beiträge des 18. Internationalen Wittgenstein Symposiums 13–20. August 1995 Kirchberg am Wechsel*, 1995. Kirchberg am Wechsel: Die Österreichische Ludwig Wittgenstein Gesellschaft. p. 770–776. <http://hdl.handle.net/1956/1875>.

[Piez (2001)] Wendell Piez. 'Beyond the 'Descriptive vs. Procedural' Distinction'. B. Tommie Usdin, Steven R. Newcomb (eds.) *Proceedings of Extreme Markup Languages 2001: Montreal, Canada*, 2001. <http://www.mulberrytech.com/Extreme/Proceedings/html/2001/Piez01/EML2001Piez01.html>. <http://www.idealliance.org/papers/extreme/proceedings/html/2001/Piez01/EML2001Piez01.html>.

[Popham (1996)] Michael Popham. 'What Is Markup and Why Does It Matter'. Michael Popham, Lorna Hughes (eds.) *Computers and Teaching in the Humanities: Selected Papers from the CATH94 Conference held in Glasgow University September 9th-12th 1994*, 1996. Oxford: CTI Centre for Textual Studies.

[Quin (1996)] Liam Quin. 'Suggestive Markup: Explicit Relationships in Descriptive and Prescriptive DTDs'. B. Tommie Usdin, Deborah A. Lapeyre (eds.) *SGML'96 Conference Proceedings*, 1996. Alexandria, VA: Graphic Communications Association. p. 405–418. <http://www.holoweb.net/~liam/papers/1996-sgml96-SuggestiveMarkup/>.

[Raymond et al. (1996)] Darrell Raymond, Frank Tompa, Derick Wood. 'From Data Representation to Data Model: Meta-Semantic Issues in the Evolution of SGML'. *Computer Standards & Interfaces* 1996. 18 (1) p. 25–36. <http://www.cs.uwaterloo.ca/~fwtompa/.papers/sgml.ps>. <http://hdl.handle.net/1783.1/41>. <http://xml.coverpages.org/raymmeta.ps>.

[Renear et al. (1996)] Allen Renear, David Durand, Elli Mylonas. 'Refining our Notion of What Text Really Is: The Problem of Overlapping Hierarchies'. Susan Hockey, Nancy Ide (eds.) *Research in Humanities Computing 4: Selected Papers from the 1992 ALLC/ACH Conference*, 1996. Oxford: Oxford University Press. p. 263–280. <http://www.stg.brown.edu/resources/stg/monographs/ohco.html>.

[Renear (1997)] Allen Renear. 'Out of Praxis: Three (Meta)Theories of Textuality'. Kathryn Sutherland (ed.) *Electronic Text: Investigations in Method and Theory*, 1997. New York, NY: Clarendon Press Oxford. p. 107–126.

[Renear (2000)] Allen Renear. 'The Descriptive/Procedural Distinction is Flawed'. *Markup Languages: Theory and Practice* 2000. 2 (4) p. 411–420.

[Renear et al. (2002)] Allen H. Renear, David Dubin, C. Michael Sperberg-McQueen. 'Towards a Semantics for XML Markup'. Richard Furuta, Jonathan I. Maletic, Ethan V. Munson (eds.) *Proceedings of the 2002 ACM Symposium on Document Engineering*, 2002. McLean, VA: Association for Computing Machinery. p. 119–126. <http://doi.acm.org/10.1145/585058.585081>.

[Renear et al. (2003)] Allen H. Renear, Christopher Phillippe, Pat Lawton, David Dubin. 'An XML Document Corresponds to Which FRBR Group 1 Entity?'. B. Tommie Usdin, Steven R. Newcomb (eds.) *Proceedings of Extreme Markup Languages 2003: Montreal, Canada*, 2003. <http://www.mulberrytech.com/Extreme/Proceedings/html/2003/Lawton01/EML2003Lawton01.html>. <http://www.idealliance.org/papers/extreme/proceedings/html/2003/Lawton01/EML2003Lawton01.html>.

[Renear et al. (2003)] Allen H. Renear, David Dubin, C. Michael Sperberg-McQueen, Claus Huitfeldt. 'XML Semantics and Digital Libraries'. *Proceedings of the 3rd ACM/IEEE–CS Joint Conference on Digital Libraries*, 2003. Los Alamitos, CA: IEEE Computer Society. p. 303–305. <http://portal.acm.org/citation.cfm?id=827192>.

[Renear (2004)] Allen H. Renear. 'Text Encoding'. Susan Schreibman, Ray Siemans, John Unsworth (eds.) *A Companion to Digital Humanities*, 2004. Oxford: Blackwell. p. 218–239. <http://www.digitalhumanities.org/companion/>.

[Salmon-Alt (2006)] Susanne Salmon-Alt. 'Data Structures for Etymology: Towards an Etymological Lexical Network'. *BULAG: revue internationale annuelle Numéro Etymologie* 2006. Besançon: Presses Universitaires de Franche-Comté. 31. <http://www.atilf.fr/perso/salmon-alt/telechargement/Bulag_2006.pdf>.

[Schreibman (2002)] Susan Schreibman. 'Computer-mediated Texts and Textuality: Theory and Practice'. *Computers and the Humanities* 2002. 36 (3) p. 283–293. <http://dx.doi.org/10.1023/A:1016178200469>.

[Schreibman (2002)] Susan Schreibman. 'The Text Ported'. *Literary and Linguistic Computing* 2002. 17 (1) p. 77–87. <http://dx.doi.org/10.1093/llc/17.1.77>.

[SGML Users' Group (1990)] SGML Users' Group. *A Brief History of the Development of SGML*, 1990. <http://www.sgmlsource.com/history/sgmlhist.htm>.

[Shipman and Marshall (1999)] Frank M. ShipmanIII, Catherine C. Marshall. 'Formality Considered Harmful: Experiences, Emerging Themes, and Directions on the Use of Formal Representations in Interactive Systems'. *Computer-Supported Cooperative Work* 1999. 8 (4) p. 333–352. <http://dx.doi.org/10.1023/A:1008716330212>. <http://www.csdl.tamu.edu/~shipman/papers/cscw.pdf>.

[Sperberg-McQueen and Huitfeld (1999)] C. Michael Sperberg-McQueen, Claus Huitfeld. 'Concurrent Document Hierarchies in MECS and SGML'. *Literary and Linguistic Computing* 1999. 14 (1) pp. 29-42.

[Sperberg-McQueen and Huitfeldt (1999)] C. Michael Sperberg-McQueen, Claus Huitfeldt. 'Concurrent document hierarchies in MECS and SGML'. *Literary and Linguistic Computing* 1999. 14 (1) p. 29–42. <http://dx.doi.org/10.1093/llc/14.1.29>.

[Sperberg-McQueen et al. (2000)] C. Michael Sperberg-McQueen, Claus Huitfeldt, Allen H. Renear. 'Meaning and Interpretation in Markup'. *Markup Languages: Theory and Practice* 2000. 2 (3) p. 215–234.

[Sperberg-McQueen et al. (2002)] C. Michael Sperberg-McQueen, David Dubin, Claus Huitfeldt, Allen Renear. 'Drawing Inferences on the Basis of Markup'. B. Tommie Usdin, Steven R. Newcomb (eds.) *Proceedings of Extreme Markup Languages 2002: Montreal, Canada*, 2002. <http://www.mulberrytech.com/Extreme/Proceedings/html/2002/CMSMcQ01/EML2002CMSMcQ01.html>. <http://www.idealliance.org/papers/extreme/proceedings/html/2002/CMSMcQ01/EML2002CMSMcQ01.html>.

[Sperberg-McQueen (2006)] C. Michael Sperberg-McQueen. 'Rabbit/duck grammars: a validation method for overlapping structures'. *Proceedings of Extreme Markup Languages 2006*, 2006. <http://www.idealliance.org/papers/extreme/proceedings/html/2006/SperbergMcQueen01/EML2006SperbergMcQueen01.html>.

[Sukovic (2002)] Suzana Sukovic. 'Beyond the Scriptorium: The Role of the Library in Text Encoding'. *D-Lib* 2002. 8 (1) . <http://www.dlib.org/dlib/january02/sukovic/01sukovic.html>.

[University of Nebraska – Lincoln Libraries (2003)] University of Nebraska – Lincoln Libraries. *A Basic Guide to Text Encoding*, 2003. <http://libr.unl.edu:2000/guide_site/teien.html>.

[Unsworth (2000)] John Unsworth. *Scholarly Primitives: What Methods Do Humanities Researchers Have in Common, How Might Our Tools Reflect This?*, 2000. (Part of a Symposium on "Humanities Computing: Formal Methods, Experimental Practice" sponsored by King's College, London) <http://www3.isrl.uiuc.edu/~unsworth/Kings.5-00/primitives.html>.

[Unsworth (2001)] John Unsworth. *Knowledge Representation in Humanities Computing*, 2001. (Lecture I in the eHumanities NEH Lecture Series on Technology & the Humanities, Washington, DC, April 3, 2001) <http://www3.isrl.uiuc.edu/~unsworth/KR/>.

[Unsworth et al. (eds.) (2004)] John Unsworth, Katherine O'BrienKatherine O'Keeffe, Lou Burnard (eds.) *Electronic Textual Editing*, 2004. TEI Consortium. <http://www.tei-c.org/Activities/ETE/>.

[Vitali et al. (2000)] Fabio Vitali, Luca Bompani, Paolo Ciancarini. 'Hypertext Functionalities with XML'. *Markup Languages: Theory & Practice* 2000. 2 (4) p. 389.

[Watson (1992)] Dennis G. Watson. *Brief History of Document Markup*, 1992. (Circular 1086. Florida Cooperative Extension Service, Institute of Food and Agricultural Sciences, University of Florida) <http://edis.ifas.ufl.edu/BODY_AE038>.

[Weel ((no date))] Adriaan van der Weel. 'The Concept of Markup'. *Digital Text and the Gutenberg Heritage*, (no date). chapter 3. <http://www.let.leidenuniv.nl/wgbw/~adriaan/Gut/Ch03_Concept_of_markup.fn.pdf>.(in preparation; draft only)

[Welty and Ide (1999)] Christopher Welty, Nancy Ide. 'Using the Right Tools: Enhancing Retrieval from Marked-up Documents'. *Computers and the Humanities* 1999. 33 (1–2) p. 59–84. <http://dx.doi.org/10.1023/A:1001800717376>. <http://www.cs.vassar.edu/faculty/welty/papers/CHUM-99.pdf>.

TEI

[An Agreement to Establish a Consortium for the Maintenance of the Text Encoding Initiative (March 1999)] *An Agreement to Establish a Consortium for the Maintenance of the Text Encoding Initiative*, March 1999. <http://www.tei-c.org/Consortium/consortium.html>.

[Bauman (1995)] Syd Bauman. 'Tables of Contents TEI-style'. Lou Burnard (ed.) *TEXT Technology: The Journal of Computer Text Processing Electronic Texts and the Text Encoding Initiative. A Special Issue of TEXT Technology* 1995. Madison, SD: College of Liberal Arts, Dakota State University. 5 (3) p. 235–247.

[Bauman (1996)] Syd Bauman. 'Keying NAMEs: the WWP Approach'. *Brown University Women Writers Project Newsletter* 1996. 2 (3) p. 3–6 p. 10–11. <http://www.wwp.brown.edu/project/newsletter/vol02num03/nameKey-home.html>.

[Bauman and Catapano (1999)] Syd Bauman, Terry Catapano. 'TEI and the Encoding of the Physical Structure of Books'. *Computers and the Humanities* 1999. 33 (1–2) p. 113–127. <http://dx.doi.org/10.1023/A:1001769103586>.

[Bauman and Flanders (2004)] Syd Bauman, Julia Flanders. 'Odd Customizations'. *Proceedings of Extreme Markup Languages 2004*, 2004. <http://www.mulberrytech.com/Extreme/Proceedings/html/2004/Bauman01/EML2004Bauman01.html>. <http://www.idealliance.org/papers/extreme/proceedings/html/2004/Bauman01/EML2004Bauman01.html>.

[Bauman (2005)] Syd Bauman. 'TEI HORSEing Around'. *Proceedings of the Extreme Markup Languages 2005*, 2005. <http://www.mulberrytech.com/Extreme/Proceedings/html/2005/Bauman01/EML2005Bauman01.html>. <http://www.idealliance.org/papers/extreme/proceedings/html/2005/Bauman01/EML2005Bauman01.html>.

[Brown (1994)] Malcolm B. Brown. 'What is the TEI?'. *Information Technology and Libraries* 1994. 13 (1) p. 8.

[Burnard (1992)] Lou Burnard. 'The Text Encoding Initiative: A Progress Report'. Gerhard Leitner (ed.) *New Directions in Corpus Linguistics*, 1992. Berlin: Mouton de Gruyter.

[Burnard (1993)] Lou Burnard. 'Rolling your own with the TEI'. *Information Services and Use* 1993. Amsterdam: IOS Press. 13 (2) p. 141–154.

[Burnard (1994)] Lou Burnard. 'The TEI: Towards an Extensible Standard for the Encoding of Texts'. Seamus Ross, Edward Higgs (eds.) *Electronic Information Resources and Historians*, 1994. London: British Academy.

[Burnard (1995)] Lou Burnard. 'The Text Encoding Initiative: An Overview'. Geoffrey Leech, Greg Myers, Jenny Thomas (eds.) *Spoken English on Computer: Transcription, Mark-up and Application*, 1995. London: Longman.

[Burnard (1997)] Lou Burnard. *The Text Encoding Initiative's Recommendations for the Encoding of Language Corpora: Theory and Practice*, 1997. <http://users.ox.ac.uk/~lou/wip/Soria/>.(Prepared for a seminar on Etiquetación y extracción de información de grandes corpus textuales within the Curso Industrias de la Lengua (14–18 de Julio de 1997). Sponsored by the Fundacion Duques de Soria.)

[Burnard and Popham (1999)] Lou Burnard, Michael Popham. 'Putting Our Headers Together: A Report on the TEI Header Meeting 12 September 1997.'. *Computers and the Humanities* 1999. Dordrecht, Boston: Kluwer Academic Publishers. 33 (1-2) p. 39–47. <http://dx.doi.org/10.1023/A:1001710828622>.

[Burnard (2000)] Lou Burnard. *Text Encoding for Interchange: A New Consortium*, 2000. <http://www.ariadne.ac.uk/issue24/tei/>.

[Chang (2001)] Sheau-Hwang Chang. 'The Implications of TEI'. *OCLC Systems and Services* 2001. 17 (3) p. 101–103.

[Ciotti (ed.) (2005)] Fabio Ciotti (ed.) *Il Manuale TEI Lite: Introduzione Alla Codifica Elettronica Dei Testi Letterari*, 2005. Milano: Sylvestre Bonnard.

[Cournane (1997)] Mavis Cournane. *The Application of SGML/TEI to the Processing of Complex, Multi-lingual Text*, (PhD Dissertation) 1997. Cork, Ireland: University College Cork.

[Digital Library Federation (1998)] Digital Library Federation. *TEI and XML in Digital Libraries: Meeting June 30 and July 1, 1998, Library of Congress, Summary/Proceedings*, 1998. <http://www.umdl.umich.edu/workshops/teidlf/>.

[Digital Library Federation (2007)] Digital Library Federation. *TEI Text Encoding in Libraries: Guidelines for Best Encoding Practices, Version 2.1 (March 27, 2006)*, 2007. <http://www.diglib.org/standards/tei.htm>.

[Finney (2006)] Timothy J. Finney. 'Manuscript Markup'. Larry W. Hurtado (ed.) *The Freer Biblical Manuscripts: Fresh Studies of an American Treasure Trove*, 2006. Atlanta, GA: Society of Biblical Literature. pp. 263-288. Text-critical studies 6.

[Gibson and Ruotolo (2003)] Matthew Gibson, Christine Ruotolo. 'Beyond the Web: TEI, the Digital Library, and the Ebook Revolution'. *Computers and the Humanities* 2003. 37 (1) p. 57–63. <http://dx.doi.org/10.1023/A:1021895322291>.

[Loiseau ((no date))] Sylvain Loiseau. *Introduction à la TEI*, (no date). <http://revue-texto.net/Corpus/Manufacture/standards/d1e284.html>.

[Marko and Kelleher Powell (2001)] Lynn Marko, Christina Kelleher Powell. 'Descriptive Metadata Strategy for TEI Headers: A University of Michigan Library Case Study'. *OCLC Systems & Services* 2001. 17 (3) pp. 117-20. <http://dx.doi.org/10.1108/10650750110402585>.

[Mertz (2003)] David Mertz. *XML Matters: TEI — the Text Encoding Initiative, An XML Dialect for Archival and Complex Documents*, 2003. <http://www-106.ibm.com/developerworks/xml/library/x-matters30.html>.

[Morrison (1999)] Alan Morrison. 'Delivering Electronic Texts Over the Web: The Current and Planned Practices of the Oxford Text Archive'. *Computers and the Humanities* 1999. 33 (1-2) pp. 193-198. <http://dx.doi.org/10.1023/A:1001726011322>.

[Mylonas and Renear (1999)] Elli Mylonas, Allen Renear. 'The Text Encoding Initiative at 10: Not Just an Interchange Format Anymore – But a New Research Community'. *Computers and the Humanities* 1999. 33 (1-2) pp. 1-9. <http://dx.doi.org/10.1023/A:1001832310939>.

[Nellhaus (2001)] Tobin Nellhaus. 'XML, TEI, Digital Libraries in the Humanities'. *Portal: Libraries and the Academy* 2001. 1 (3) pp. 267-277. <http://muse.jhu.edu/journals/portal_libraries_and_the_academy/v001/1.3nellhaus.html>.

[Rahtz (2003)] Sebastian Rahtz. *Building TEI DTDs and Schemas on demand*, 2003. (Paper presented at XML Europe 2003, London, March 2003) <http://www.idealliance.org/papers/dx_xmle03/papers/03-01-04/03-01-04.html>.

[Rahtz et al. (2004)] Sebastian Rahtz, Norman Walsh, Lou Burnard. *A unified model for text markup: TEI, Docbook, and beyond*, 2004. (Paper presented at XML Europe 2004, Amsterdam, April 2004) <http://www.idealliance.org/papers/dx_xmle04/papers/03-08-01/03-08-01.html>.

[Renear (1995)] Allen Renear. 'Theory and Metatheory in the Development of Text Encoding'. Michael A. R. Biggs, Claus Huitfeldt (eds.) *Philosophy and Electronic Publishing*, 1995. (Interactive seminar for the Monist) <http://hhobel.phl.univie.ac.at/mii/pesp.html>.

[Robinson ((no date))] Peter Robinson. *Making a Digital Edition with TEI and Anastasia*, (no date). <http://www.cta.dmu.ac.uk:8000/AnaServer?teidoc+0+start.anv>.

[Seaman (1995)] David Seaman. *The Electronic Text Center Introduction to TEI and Guide to Document Preparation*, 1995. <http://etext.lib.virginia.edu/tei/uvatei.html>.

[Simons (1999)] Gary F. Simons. 'Using Architectural Forms to Map TEI Data into an Object-Oriented Database'. *Computers and the Humanities* 1999. 33 (1-2) pp. 85-101. <http://dx.doi.org/10.1023/A:1001765030032>.

[Smith (1999)] David Smith. 'Textual Variation and Version Control in the TEI'. *Computers and the Humanities* 1999. 33 (1-2) pp. 103-112. <http://dx.doi.org/10.1023/A:1001795210724>.

[Sperberg-McQueen (1991)] C. Michael Sperberg-McQueen. 'Text in the Electronic Age: Textual Study and Text Encoding, with Examples from Medieval Texts'. *Literary & Linguistic Computing* 1991. 6 (1) pp. 34-46. <http://dx.doi.org/10.1093/llc/6.1.34>.

[Sperberg-McQueen (1994)] C. Michael Sperberg-McQueen. 'The Text Encoding Initiative: Electronic Text Markup for Research'. Brett Sutton (ed.) *Literary Texts in an Electronic Age*, 1994. Urbana-Champaign, IL: University of Illinois at Urbana-Champaign, Graduate School of Library and Information Science. p. 35–55.

[Sperberg-McQueen (1996)] C. Michael Sperberg-McQueen. 'Textual Criticism and the Text Encoding Initiative'. Richard J. Finneran (ed.) *The Literary Text in the Digital Age*, 1996. Ann Arbor, MI: University of Michigan Press. p. 37–62.

[Vanhoutte (2004)] Edward Vanhoutte. 'An Introduction to the TEI and the TEI Consortium'. *Literary & Linguistic Computing* 2004. 19 (1) p. 9. <http://dx.doi.org/10.1093/llc/19.1.9>.

Appendix G

Prefatory Notes

This Appendix contains (in reverse chronological order) the 'Introductory Notes' prefixed to each revision of the TEI Guidelines since its first publication in 1994.

Prefatory Note (March 2002)

The primary goal of this revision has been to make available a new and corrected version of the TEI Guidelines which:

- is expressed in XML and conforms to a TEI-conformant XML DTD;
- generates a set of DTD fragments that can be combined together to form either SGML or XML document type definitions;
- corrects blatant errors, typographical mishaps, and other egregious editorial oversights;
- can be processed and maintained using readily available XML tools instead of the special-purpose ad hoc software originally used for TEI P3.

A second major design goal of this revision has been to ensure that the DTD fragments generated would not break existing documents: in other words, that any document conforming to the original TEI P3 SGML DTD would also conform to the new XML version of it. Although full backwards compatibility cannot be guaranteed, we believe our implementation is consistent with that goal.

In most respects, the TEI Guidelines have stood the test of time remarkably well. The present edition makes no substantial attempt to rewrite those few parts of them which have now been rendered obsolete by changes since their first publication, though an indication is given in the text of where such rewriting is now considered necessary. Neither does the present version attempt to address any of the many possible new areas of digital activity in which the TEI approach to standardization may have something to offer. Both these tasks require the existence of an informed and active TEI Council to direct and validate such extension and maintenance work, in response to the changing needs and priorities of the TEI user community.

Two exceptions to the above principles may be cited: firstly, the chapter which originally provided a 'Gentle Introduction' to SGML has been completely rewritten to provide a similarly gentle introduction to XML; secondly the chapter on character sets has been completely revised in light of the close connexion between Unicode and XML. The editors gratefully acknowledge the assistance of the *ad hoc* workgroup chaired by Christian Wittern, which undertook to provide expert advice and correction at very short notice, in the latter task.

The preparation of this new version relied extensively on preliminary work carried out by the former North American editor of the TEI Guidelines, C.M. Sperberg-McQueen. In a TEI working paper written in 1999[1] he

[1] *TEI ED W69*, available from the TEI web site at `http://www.tei-c.org/Vault/ED/edw69.htm`.

sketched out a precise blueprint for the conversion of the TEI from SGML to XML, which we have implemented, with only slight modification.

The Editors would also like to express thanks to the team of volunteers from the TEI community who helped us with the task of proofreading the first draft during the summer of 2001; and to Sebastian Rahtz of Oxford University Computing Services, without whose skill and enthusiasm this new edition would not have been possible.

A substantial proportion of the work of preparing this new edition was funded with the assistance of a grant from the US National Endowment for the Humanities, whose continued support of the TEI has also been crucial to the effort of setting up the TEI Consortium.

Finally, we would like to thank all our colleagues on the interim management board of the TEI Consortium, in particular its Chairman John Unsworth, for their continued support of the TEI's work, and their willingness to devote effort to the difficult task of overseeing its transition to a new organizational infrastructure.

Summary details of the changes made in the present and previous editions are given in their Prefatory Notes, all of which are now reproduced in an Appendix to the present edition: see *Appendix G Prefatory Notes*.
Lou Burnard and Syd Bauman (TEI Editors) Oxford and Providence, March 2002.

Introductory Note (November 2001)

To complete the work started in June of this year, the TEI Editors asked for volunteers from the TEI community to proofread the preliminary XML version. 24 volunteers responded to this call during August, and gave invaluable help both by identifying a number of previously un-noticed errors, and by suggesting areas in which more substantial revision should be undertaken in the future. The Editors gratefully acknowledge the assistance of the following individuals during this exercise:

> Jimmy Adair, Syd Bauman, Michael Beddow, Steven Bird, Lisa Charlong, Matthew Driscoll, Patrick Durusau, Tomaz Erjavec, Nick Finke, Tim Finney, Julia Flanders, Mike Fraser, Pankaj Kamthan, François Lachance, Terry Langendoen, Anne Mahoney, Gregory Murphy, Daniel Pitti, Rafal Prinke, Laurent Romary, Stewart Russell, Gary Simons, Elisabeth Solopova, Christian Wittern, Martin Wynne.

In addition to error correction, and clear delineation of those sections in which substantial revision is yet to be undertaken for TEI P5, the present draft differs from earlier ones in the following respects:

- Formal Public Identifiers have been introduced as a means of constructing TEI DTDs and an SGML Open Catalog is now included with the standard release;
- Some systematic errors and omissions in the reference section have been removed; the format of this section has been substantially changed, we hope for the better;
- The chapters on obtaining the TEI DTDs and WSDs have been brought up to date; the chapter on modification has been expanded to include a discussion of the TEI Lite customization;
- All examples and cited markup has been checked for XML validity against the published DTDs, and corrected where faulty; examples have been formatted in a (more or less) consistent style.

Lou Burnard and Syd Bauman (Editors)Oxford and Providence, November 2001.

Introductory Note (June 2001)

This is a preliminary version of a revised and fully XML-compliant edition of the TEI Guidelines. Although work on revising and correcting the text of the document is incomplete, by making available this preliminary version we hope to facilitate testing of the XML document type declarations which it describes by as wide a range of TEI users as possible.

The primary goal of this revision is to make available the corrected (May 1999) edition of the Guidelines in a new version which:

- is expressed in XML and itself conforms to a TEI-conformant XML DTD;
- generates a set of XML DTD fragments that can be combined together in the same way as the existing TEI (P3) SGML DTD fragments to form true TEI XML DTD fragments without loss of functionality;
- can be processed and maintained using readily available XML tools instead of the special-purpose ad hoc software originally used for TEI P3.

As noted elsewhere, a number of errors were corrected in the May 1999 edition. A (much) smaller number of errors have also been corrected in this edition, but no new material has been added. We expect the expansion and modification of the Guidelines to become a real possibility in the context of the newly formed TEI Consortium, which has funded the preparation of this present edition.

A major design goal of both this and the previous revision has been to ensure that the DTD fragments generated would not break existing documents: in other words, that any document conforming to the original TEI P3 SGML DTD would also conform to the new XML version of it. Although full backwards compatibility cannot be guaranteed, we believe our implementation is consistent with that goal.

In making this new version, we relied extensively on preliminary work carried out by the outgoing North American editor of the TEI Guidelines, Michael Sperberg-McQueen. In a TEI working paper written in 1999, TEI ED W69, Michael sketched out a precise blueprint for the conversion of the TEI from SGML to XML, which we have implemented, with only slight modification. The current TEI editors wish to express here our admiration for the detailed care put into that paper, without which our task would have been forbiddingly difficult, if not impossible. We would also like to express our thanks to Sebastian Rahtz of Oxford University Computing Services, for his invaluable assistance in preparing this new edition.

We list here in summary form all the changes made in the present edition. Full technical details are provided in documents TEI EDW69 and TEI EDW70, available from the TEI web site.

1. A new keyword TEI.XML has been added. By setting its value to **INCLUDE**, rather than the default **IGNORE**, the user can request generation of an XML rather than an SGML DTD;

2. The content models of all elements have been checked, and, where necessary, changed so that they are equally valid as SGML or as XML;

3. The declared value for all attributes has been changed to a form which is equally valid as SGML or as XML;

4. All the examples have been checked for conformance and converted to use XML syntax, where possible. (This process is currently incomplete.)

5. Some errors and duplications in the class membership of elements from the names and dates tagsets have been corrected.

To implement the first of these, we have parameterized the tag omissibility indicators '- o' and '- -' used within element declarations in the DTD. When XML is to be generated, the parameter entities concerned are redeclared with the null string as their value.

The second change was achieved by removing SGML-specific features (ampersand connectors, inclusion and exclusion exceptions, various types of attribute content) from the DTD and revising the syntax of the DTD to conform to XML requirements (specifically in the representation of mixed-content models, and by removing redundant parentheses). In making these changes, we took care to ensure that the resulting content model would continue to accept existing valid documents, though in the nature of things it could not be guaranteed to reject the same set of documents. As further discussed in EDW69 and EDW70, some constraints (exclusion exceptions, for example) which could be carried out by a generic SGML parser using TEI P3 will have to be implemented by a special purpose TEI validator using TEI P4.

Much work remains to be done, firstly in testing the new DTD fragments against as wide a range of TEI materials as possible, secondly in revising the discussion of markup theory and practice within the text to reflect current thinking. A few sections of the current text (the Gentle Introduction to SGML and the discussion of Extended Pointer syntax are two examples) will need substantial rewriting. For the most part, however, we think the Guidelines have stood the test of time well and can be recommended to a new generation of text encoders scarcely born at the time they were first formulated.

Lou Burnard and Steve De Rose (Editors)

Oxford and Providence, May 2001.

Introductory Note (May 1999)

No work of the size and complexity of the TEI *Guidelines* could reasonably be expected to be error-free on publication, nor to remain long uncorrected. It has however taken rather longer than might have been anticipated to complete production of the present corrected reprint of the first edition, for which we present our apologies, both to the many individuals and institutions whose enthusiastic adoption and promotion of the TEI encoding scheme have ensured its continued survival in the rapidly changing world of digital scholarship, and also to the many helpfully critical users whose assiduous uncovering and reporting of our errors have made possible the present revision.

At its first meeting in Bergen, in June 1996, the TEI Technical Review Committee (TRC) approved the setting up of a small working committee to oversee the production of a revised edition of the TEI *Guidelines*, to include corrections of as many as possible of the `corrigible errors' notified to the editors since publication of the first edition in May 1994, the bulk of which are summarized in a TEI working paper (TEI EDW67, available from the TEI web site).

During the spring of 1997, this TRC Core Subcommittee reviewed nearly 200 comments and proposals which the editors had collected from public debate and discussion over the preceding two years, and provided invaluable technical guidance in disposition of them. We are glad to take this opportunity of expressing our thanks to this subcommittee, whose members were Elli Mylonas, Dominic Dunlop, and David T. Barnard.

The work of making the corrections and regenerating the text proceeded rather fitfully during 1998 and 1999, largely because of increasing demands on the editors' time from their other responsibilities. With the establishment of the new TEI Consortium, it is be hoped that maintenance of the Guidelines will be placed on a more secure footing. Some specific areas in which we anticipate future revisions being carried out are listed below.

Typographic corrections made

- examples of TEI markup throughout the text were all checked against the relevant DTD fragment and an embarassingly large number of tagging errors corrected;
- various minor typographic and spelling errors were corrected;
- the 'corrigible errors' listed in working paper TEI EDW67 were all corrected: some of these required specific changes to the DTD which are listed in the next section.

Specific changes in the DTD

A major goal of this revision was to avoid changes which might invalidate existing data, even where existing constructs seemed erroneous in retrospect. To that end, wherever changes have been made in content models for existing elements, they have as far as possible been made so that the DTD will now accept a superset of what was previously legal. Only one new element (<ab>) has been added.

Where possible, a few content models have been changed in such a way as to facilitate conversion to XML, but XML compatibility is *not* a goal of this revision.

Brief details of all changes made in the DTD follow:

- Several changes were made in class membership, in order to correct unreachability problems. Specifically:
  - elements <geogName>, <persName>, <placeName> were added to the m.data class;
  - <geogName> and <placeName> were removed from the m.placepart class;
  - the elements <addSpan>, <delSpan>, <gap>, were added to the m.Edit class;
  - a new class m.editIncl was defined, with members <addSpan>, <delSpan>, and <gap>; this class was then added to the global inclusion class m.globIncl along with <anchor> (erroneously a member of the m.Seg class, from which it is now removed), m.metadata and m.refsys;
- added <name> element to m.addrPart class;
- added <dateline> to m.divtop and m.divbot classes;
- added <epilogue> and <castList> to m.dramafront class;
- added <divGen> to m.front class;
- added <dateline> to m.divtop and m.divtop classes;
- added <u> element to a.declaring class;
- defined new class m.fmchunk (front matter chunk), comprising <argument>, <byline>, <docAuthor>, <docDate>, <docEdition>, <docImprint>, <docTitle>, <epigraph>, <head>, and <titlePart> for use in simplification of the content model for <front> element;
- defined new element <ab> (anonymous block), and added it to the m.chunk class;
- corrected an error whereby global attributes were not properly defined for elements specifying a non-default value for any of the a.global attributes: elements affected include: <foreign>, <hi>, <del>, <pb>, <lb>, <cb>, <language>, <anchor>, and <when>;
- changed content models to permit empty <list> and empty <availability> elements;
- changed content model for <series> element to permit #PCDATA;
- changed content model for <setting> element to permit <date> element as a direct child;
- added a **key** attribute to the <distance> element, for consistency with other elements in its class;
- changed content model for <orgName> element to make it more consistent with e.g. <persName>;
- changed content model for <opener> element to include <argument>, <byline>, and <epigraph>;
- changed content models for <app>, <rdgGrp>, and <wit> elements;
- revised attributes on <hand> element.

A number of content models were changed with a view to easing the creation of an XML compatible version of the Guidelines. Specifically:

- removed ampersand connectors from <cit>, <respStmt>, <publicationStmt>, and <graph>;
- changed the mixed content models for <sense>, <re>, <persName>, <placeName>, <geogName>, <dateStruct>, <timeStruct>, and <dateline> to make them XML-conformant.

Outstanding errors

A small number of other known problems remain uncorrected in this version and are briefly listed below. Please watch the TEI mailing list for announcements of their correction.

- elements of class inter don't always behave as they should (e.g. one cannot insert a <table> before anything else in a);

- some mixed-content problems consequent on the definition of macro.specialPara need to be addressed systematically; in particular, the treatment of list items or notes which contain several paragraphs continues to surprise many users: no whitespace is allowed between the paragraphs;
- the **resp** attributes on editorial elements are not consistently defined;
- the discussions of DTD invocation, and the DTD itself, all use system identifiers instead of formal public identifiers.

Our next priority however will be the production of a fully XML-compliant version of the TEI DTD, work on which is already well advanced.C.M. Sperberg McQueen and Lou Burnard, May 1999

Preface (April 1994)

These Guidelines are the result of over five years' effort by members of the research and academic community within the framework of an international cooperative project called the Text Encoding Initiative (TEI), established in 1987 under the joint sponsorship of the Association for Computers and the Humanities, the Association for Computational Linguistics, and the Association for Literary and Linguistic Computing.

The impetus for the project came from the humanities computing community, which sought a common encoding scheme for complex textual structures in order to reduce the diversity of existing encoding practices, simplify processing by machine, and encourage the sharing of electronic texts. It soon became apparent that a sufficiently flexible scheme could provide solutions for text encoding problems generally. The scope of the TEI was therefore broadened to meet the varied encoding requirements of any discipline or application. Thus, the TEI became the only systematized attempt to develop a fully general text encoding model and set of encoding conventions based upon it, suitable for processing and analysis of any type of text, in any language, and intended to serve the increasing range of existing (and potential) applications and use.

What is published here is a major milestone in this effort. It provides a single, coherent framework for all kinds of text encoding which is hardware-, software- and application-independent. Within this framework, it specifies encoding conventions for a number of key text types and features. The ongoing work of the TEI is to extend the scheme presented here to cover additional text types and features, as well as to continue to refine its encoding recommendations on the basis of extensive experience with their actual application and use.

We therefore offer these Guidelines to the user community for use in the same spirit of active collaboration and cooperation with which they have so far been developed. The TEI is committed to actively supporting the wide-spread and large-scale use of the Guidelines which, with the publication of this volume, is now for the first time possible. In addition, we anticipate that users of the TEI Guidelines will in some instances adapt and extend them as necessary to suit particular needs; we invite such users to engage in the further development of the Guidelines by working with us as they do so.

Like any standard which is actually used, these Guidelines do not represent a static finished work, but rather one which will evolve over time with the active involvement of its community of users. We invite and encourage the participation of the user community in this process, in order to ensure that the TEI Guidelines become and remain useful in all sorts of work with machine-readable texts.

This document was made possible in part by financial support from the U.S. National Endowment for the Humanities, an independent federal agency; Directorate General XIII of the Commission of the European Communities; the Andrew W. Mellon Foundation; and the Social Science and Humanities Research Council of Canada. Direct and indirect support has also been received from the University of Illinois at Chicago, the Oxford University Computing Services, the University of Arizona, the University of Oslo and Queen's University (Kingston, Ont.), and Ohio State University.

The production of this document has been greatly facilitated by the willingness of many software vendors to provide us with evaluation versions of their products. Most parts of this text have been processed at some time by almost every currently available SGML-aware software system. In particular, we gratefully acknowledge the assistance of the following vendors:

- Berger-Levrault AIS s.a. (for Balise);
- E2S n.v. (for E2S Advanced SGML Editor);
- Electronic Book Technology (for DynaText);
- SEMA Group and Yard Software (for Mark-It and Write-It);
- Software Exoterica (for CheckMark and Xtran);
- SoftQuad, Inc., (for Author/Editor and RulesBuilder);
- WordPerfect Corporation (for Intellitag);
- Xerox Corporation (for Ventura Publisher).

Details of the software actually used to produce the current document are given in the colophon at the end of the work.

Acknowledgments

Many people have given of their time, energy, expertise, and support in the creation of this document; it is unfortunately not possible to thank them all adequately. Below are listed those who have served as formal members of the TEI's Work Groups and Working Committees during its six-year history; others not so officially enfranchised also contributed much to the quality of the result.

The editors take this opportunity to acknowledge our debt to those who have patiently endured and corrected our misunderstandings of their work; we hope that they will feel the wait has not been in vain. For any errors and inconsistencies remaining, we must accept responsibility; any virtue in what is here presented, we gladly ascribe to the energies of the keen intellects listed below.

C. M. Sperberg McQueen and Lou Burnard

TEI Working Committees (1990-1993)

(Not all members listed were able to serve throughout the development of the Guidelines.)

Committee on Text Documentation: Chair: Dominik Wujastyk (Wellcome Institute for the History of Medicine)

Members 1990–1992: J. D. Byrum (Library of Congress); Marianne Gaunt (Rutgers University); Richard Giordano (Manchester University); Barbara Ann Kipfer (Independent Consultant); Hans Jørgen Marker (Danish Data Archive, Odense); Marcia Taylor (University of Essex);

Committee on Text Representation Chair: Stig Johansson (University of Oslo)

Members 1990–1992: Roberto Cencioni (Commission of the European Communities); David R. Chesnutt (University of South Carolina); Robin C. Cover (Dallas Theological Seminary); Steven J. DeRose (Electronic Book Technology Inc); David G. Durand (Boston University); Susan M. Hockey (Oxford University Computing Service); Claus Huitfeldt (University of Bergen); Francisco Marcos-Marin (University Madrid); Elli Mylonas (Harvard University); Wilhelm Ott (University of Tübingen); Allen H. Renear (Brown University); Manfred Thaller (Max-Planck-Institut für Geschichte, Göttingen)

Committee on Text Analysis and Interpretation Chair: D. Terence Langendoen (University of Arizona)

Members 1990–1992: Robert Amsler (Bell Communications Research); Stephen Anderson (Johns Hopkins University); Branimir Boguraev (IBM T. J. Watson Research Center); Nicoletta Calzolari (University of Pisa); Robert Ingria (Bolt Beranek Newman Inc); Winfried Lenders (University of Bonn); Mitch Marcus (University of Pennsylvania); Nelleke Oostdijk (University of Nijmegen); William Poser (Stanford University); Beatrice Santorini (University of Pennsylvania); Gary Simons (Summer Institute of Linguistics); Antonio Zampolli, University of Pisa.

Committee on Metalanguage and Syntax Chair: David T. Barnard (Queen's University)

Members 1990–1994: David G. Durand (Boston University); Jean-Pierre Gaspart (Associated Consultants and Software Engineers sa/nv); Nancy M. Ide (Vassar College); Lynne A. Price (Software Exoterica / Xerox PARC); Frank Tompa (University of Waterloo); Giovanni Battista Varile (Commission of the European Communities).

In addition, the two TEI editors served ex officio on each committee.

Following publication of the first draft of the TEI Guidelines (P1) in November 1990, a number of specialist work groups were charged with responsibility for drafting revisions and extensions, which, together with material already presented in P1, constitute the basis of the present work.

In addition, many members of the work groups listed below met on three occasions to review the emerging proposals in detail at technical review meetings convened by the TEI Steering Committee. These meetings, held in Myrdal, Norway (November 1991), Chicago (May 1992) and Oxford (May 1993), were largely responsible for the technical content and organization of the present work. Attendants at these meetings are starred in the list below.

TR1 Character sets Chair: Harry Gaylord* (University of Groningen); Syun Tutiya* (Chiba University).

TR2 Text criticism Chair: Peter Robinson* (Oxford University); David Chesnutt* (University of South Carolina); Robin Cover* (Dallas Theological Seminary); Robert Kraft (University of Pennsylvania); Peter Shillingsburg (Mississippi State University).

TR3 Hypertext and hypermedia Chair: Steven J. DeRose* (Electronic Book Technologies Inc); David Durand (Boston University); Edward A. Fox (Virginia State University); Eve Wilson (University of Kent).

TR4 Formulæ, Tables, figures, and graphics Chair: Paul Ellison* (University of Exeter); Anders Berglund (Independent Consultant); Dale Waldt (Thompson Professional Publishing).

TR6 Language corpora Chair: Douglas Biber* (University of Northern Arizona); Jeremy Clear (Birmingham University); Gunnel Engwall (University of Stockholm).

TR9 Manuscripts and codicology Chair: Claus Huitfeldt* (University of Bergen); Dino Buzzetti (University of Bologna); Jacqueline Hamesse (University of Louvain); Mary Keeler (Georgetown University); Christian Kloesel (Indiana University); Allen Renear* (Brown University); Donald Spaeth (Glasgow University).

TR10 Verse Chair: David Robey* (University of Manchester); Elaine Brennan* (Brown University); David Chisholm (University of Arizona); Willard McCarty (University of Toronto).

TR11 Drama and performance texts Chair: Elli Mylonas* (Harvard University); John Lavagnino* (Brandeis University); Rosanne Potter (University of Iowa).

TR12 Literary prose Chair Thomas N. Corns* (University of Wales); Christian Delcourt (University of Liège).

AI1 Linguistic description Chair: D. Terence Langendoen* (University of Arizona); Stephen R. Anderson (Johns Hopkins University); Nicoletta Calzolari (University of Pisa); Geoffrey Sampson* (University of Sussex); Gary Simons* (Summer Institute of Linguistics).

AI2 Spoken text Chair: Stig Johansson* (University of Oslo); Jane Edwards (University of California at Berkeley); Andrew Rosta (University College London).

AI3 Literary studies Chair: Paul Fortier* (University of Manitoba); Christian Delcourt (University of Liège;); Ian Lancashire (University of Toronto); Rosanne Potter (University of Iowa); David Robey* (University of Manchester).

AI4 Historical studies Chair: Daniel Greenstein* (University of Glasgow); Peter Denley (Queen Mary Westfield College, London); Ingo Kropac (University of Graz); Hans Jørgen Marker (Danish Data Archive, Odense); Jan Oldervoll (University of Tromsø); Kevin Schurer (University of Cambridge); Donald Spaeth (Glasgow University); Manfred Thaller (Max-Planck-Institut für Geschichte, Göttingen).[2]

AI5 Print dictionaries Chairs: Robert Amsler* (Bell Communications Research) and Nicoletta Calzolari (University of Pisa); Susan Armstrong-Warwick (University of Geneva); John Fought (University of Pennsylvania); Louise Guthrie (University of New Mexico); Nancy M. Ide* (Vassar College); Frank Tompa (University of Waterloo); Carol Van Ess-Dykema (US Department of Defense); Jean Veronis (University of Aix-en-Provence).

AI6 Machine lexica Chair: Robert Ingria* (Bolt Beranek Newman Inc); Susan Armstrong-Warwick (University of Geneva); Nicoletta Calzolari (University of Pisa).

AI7 Terminological data Chair: Alan Melby* (Brigham Young University) Gerhard Budin (University of Vienna); Gregory Shreve (Kent State University); Richard Strehlow (Oak Ridge National Laboratory); Sue Ellen Wright (Kent State University).

Advisory Board

Members of the TEI Advisory Board during the lifetime of the project are listed below, grouped under the name of the organization represented.

American Anthropological Association: Chad McDaniel (University of Maryland).

American Historical Association: Elizabeth A. R. Brown (Brooklyn College, CUNY).

American Philological Association: Jocelyn Penny Small (Rutgers University).

American Philosophical Association: Allen Renear (Brown University).

American Society for Information Science: Clifford A. Lynch (University of California).

Association for Computing Machinery, Special Interest Group for Information Retrieval: 1989–93: Scott Deerwester (University of Chicago); 1993- : Martha Evens (Illinois Institute of Technology).

Association for Documentary Editing: David Chesnutt (University of South Carolina).

Association for History and Computing: 1989–91: Manfred Thaller, Max-Planck-Institut für Geschichte, Göttingen; 1991- : Daniel Greenstein (Glasgow University).

Association Internationale Bible et Informatique 1989–93: Wilhelm Ott (University of Tübingen); 1993- : Winfried Bader (University of Tübingen).

Canadian Linguistic Association: Anne-Maria di Sciullo (Université du Québec à Montréal)

Dictionary Society of North America: Barbara Ann Kipfer (Independent Consultant).

[2] This Workgroup was jointly sponsored by the Association for History and Computing.

AAP Electronic Publishing Special Interest Group: 1989–92: Betsy Kiser (OCLC); 1992- : Deborah Bendig and Andrea Keyhani (OCLC).

International Federation of Library Associations and Institutions: J. D. Byrum Jr. (The Library of Congress).

Linguistic Society of America: Stephen Anderson (The Johns Hopkins University)

Modern Language Association: Randall Jones (Brigham Young University) and Ian Lancashire (University of Toronto).

Steering Committee Membership

Members of the Steering Committee of the TEI during the preparation of this work were:

Association for Computational Linguistics:

- 1987–1993: Robert A. Amsler (Bell Communications Research);
- 1987–1993: Donald E. Walker (Bell Communications Research);
- 1993–1994: Susan Armstrong-Warwick (University of Geneva);
- 1994–1999: Judith Klavans (Columbia University).

Association for Computers and the Humanities:

- 1987–1999: Nancy M. Ide (Vassar College);
- 1987–1994: C. M. Sperberg-McQueen (University of Illinois at Chicago);
- 1994–1999: David Barnard (Queen's University).

Association for Literary and Linguistic Computing:

- 1987–1999: Susan M. Hockey (Center for Electronic Texts in the Humanities);
- 1987–1999: Antonio Zampolli (University of Pisa).

Appendix H

Colophon

The text of this manual was prepared electronically on a variety of systems. Most sections were originally drafted by members of the work groups and working committees of the TEI; all have been revised by the editors to achieve greater uniformity of style and greater consistency in the tag set.

The web release of the Guidelines was created using a library of XSLT stylesheets to convert to XHTML; the PDF version for printing was produced by conversion to LaTeX markup, processed using XeLaTeX. The XSLT libraries were written by Sebastian Rahtz.

Almost every available SGML and XML editor or processing program has been used at one time or another by the TEI; but without the open source implementations of XML parsers, editors and XSLT engines by James Clark, Richard Stallman, Michael Kay, and Daniel Veillard, the TEI could not survive, and we thank these individuals. We would also like to thank the staff at Syncrosoft, creators of the oXygen editor, for their support for the TEI during the creation on P5.

Many volunteers contributed to the preparation of this release of the Guidelines; we particularly note the work of Sabine Krott, Eva Radermacher and Arianna Ciula for their work in structuring the bibliographies.

The production and release process for TEI P5 was managed by Sebastian Rahtz for the TEI Technical Council.

Index

Entries in italic font refer to examples, with those in bold italics referring to examples where the indexed element is the root.

<ab>, *445*, *489, 490*, *494*, *498*, *508*, *607*, ***739***, *813*
<abbr>, *87, 88*, *343–345*, *624*, ***741***, *758*, *813*, *887*, *889*
@absolute
  <when>, *241–243*, *501*, *1187*
<accMat>, *323*, ***742***
<acquisition>, *294*, *324, 325*, ***743***, *947*
@active
  <interaction>, *456*, *961*, *1183*
  <relation>, *413, 414*, *425, 426*, *1098, 1099*
<activity>, *459*, ***744***, *1130*
<actor>, *203*, *205–207*, ***744***, *790–792*, *1054*
<add>, *77, 78*, *132*, *351–357*, *360*, *366*, ***746***, *1159*
<additional>, *326*, ***748***
<additions>, *321*, ***749***
<addName>, *397*, *399*, ***747***
<address>, *27*, *51*, *81–83*, *420*, *750*, ***751***, *1050*, *1176*
<addrLine>, *27*, *51*, *81*, *420*, ***750***, *751*, *1176*
<addSpan>, *354*, ***747***
@adj
  <node>, *575*
@adjFrom
  <node>, *577*
@adjTo
  <node>, *577*
<adminInfo>, *326*, *748*, ***752***
<affiliation>, ***753***
<age>, ***754***
@age
  <person>, *458*, *992*, *1050*, *1057*
@agent
  <damage>, *364*, *830*
  <damageSpan>, *363, 364*
  <gap>, *364*
@aloud
  <said>, *65*, *1117*
<alt>, *513, 514*, ***755***, *756*
<altGrp>, *513, 514*, ***756***
<altIdent>, *70*, *630*, *651*, ***756***, *1010*, *1210*
<altIdentifier>, *292–294*, *304, 305*, *329*, ***757***, *1020*
<am>, *309*, *344*, *389*, ***758***
@ana
  <anchor>, *530*
  <c>, *533*, *554*
  <cl>, *533*
  <date>, *431*
  <join>, *530*
  <m>, *533*
  <phr>, *533*, *554*
  <s>, *530*
  <seg>, *530*
  <w>, *532, 533*, *554*
<analytic>, *113*, *116, 117*, *122, 123*, ***758***, *989*, *1011*
<anchor>, *98*, *241–245*, *288*, *354, 355*, *364*, *385*, *387*, *486*, *492*, *499*, *529, 530*, *608*, *610*, *622*, *747*, ***759***, *801*, *831*, *841*, *1145*
@anchored
  <note>, *469*, *471–473*, *1030*
<app>, *350*, *357*, *372–374*, *376–379*, *383–389*, ***760***, *972*, *983*, *1084*, *1220, 1221*
<appInfo>, *44*, ***761***, *762*
<application>, *44*, *761*, ***762***
<arc>, *574*, *576–578*, *580*, *582*, *586*, ***763***, *935*
<argument>, *148*, ***764***
@arity
  <tree>, *588, 589*, *591, 592*, *1196*
@assertedValue
  <certainty>, *621, 622*, *624*, *800, 801*
<att>, *40*, *49*, *51*, *109*, *628*, *633*, ***764***, *888*, *1081*, *1094*, *1176*
<attDef>, *70*, *213*, *408*, *638, 639*, *653*, *655, 656*, *666*, *675*, *677*, ***765***, *766*, *875*, *885*
<attList>, *70*, *213*, *408*, *638, 639*, *653*, *655, 656*, *666*, *675*, ***766***, *875*, *885*
<attRef>, ***766***
@atts
  <specDesc>, *1147*, *1150*

<author>, *23*, *31*, *49*, *51*, *66*, *113, 114*, *116, 117*, *119*, *121–125*, *228, 229*, *270*, *294*, *299*, *310, 311*, *326*, *381*, *718*, *758*, ***767***, *771, 772*, *774*, *782*, *989*, *997*, *1011*, *1016*, *1019*, *1042*, *1089*, *1097*, *1121*, *1126*, *1176*
<authority>, *27*, ***768***
<availability>, *27*, *327*, *752*, ***769***, *1076*, *1176*

<back>, *100*, *136*, *152*, *155*, *166, 167*, *202*, *500*, ***770***, *861*, *1182*
@baseForm
<m>, *526*, *998*
<bibl>, *31*, *43*, *50, 51*, *66*, *90*, *100*, *112–114*, *118, 119*, *125*, *149*, *157*, *164*, *228, 229*, *236*, *270*, *299*, *307*, *312*, *323*, *326*, *328*, *347*, *381*, *412*, *469*, *471–473*, *718*, ***771***, *782*, *808*, *810*, *813*, *861*, *881*, *962*, *989*, *997*, *1015*, *1019*, *1042*, *1089*, *1121*, *1126*, *1167*, *1173*, *1176*
<biblFull>, *113*, ***772***
<biblScope>, *113, 114*, *116–119*, *122, 123*, *125*, *307*, *326*, *758*, ***774***, *997*, *1011*, *1015*, *1127*
<biblStruct>, *31*, *51*, *113*, *116, 117*, *119–124*, *758*, ***774***, *989*, *1011*, *1097*
<bicond>, *566, 567*, ***775***, *914*, *954*
<binary>, *538*, *540*, *542*, *545*, *563, 564*, *566, 567*, *775*, ***776***, *820*, *892, 893*, *954*, *1185*, *1205*, *1208*
<binaryObject>, ***777***
<binding>, *296, 297*, *322*, ***777, 778***
<bindingDesc>, *322*, *778*, ***778***
<birth>, *300*, *393*, *458*, ***779***, *1057*
<bloc>, *403*, *419*, ***780***
<body>, *57*, *105*, *107*, *136*, *138*, *140–142*, *151, 152*, *155*, *158*, *183–186*, *202*, *208*, *218*, *220*, *244*, *252, 253*, *354*, *384*, *463*, *476*, *508*, *512*, *517*, *558, 559*, *737*, *747*, ***782***, *851*, *903*, *1181, 1182*
<broadcast>, *229, 230*, ***782***, *1089*
<byline>, *145*, *151, 152*, *164*, ***783, 784***, *862*, *1190*

<c>, *527*, *533*, *554*, *556*, ***784***
<caesura>, *189*, ***786***
@calendar
<date>, *710*, *832*
<camera>, *222–224*, ***787***, *788*, *1213*
<caption>, *224*, ***788***
<case>, ***789***
<castGroup>, *206, 207*, ***790***, *792*
<castItem>, *200–203*, *205–207*, *210, 211*, *214*, *744*, *790*, ***791***, *792*, *1014*, *1054, 1055*
<castList>, *200–203*, *205*, *207*, *210, 211*, *214*, ***792***, *1014*, *1054, 1055*
<catchwords>, *301*, *817*
<catDesc>, *43*, *312*, ***794***, *795, 796*, *1173*, *1182*
<category>, *43*, *312*, *795*, ***796***, *1173*, *1182*
<catRef>, *44*, *48*, ***795***, *1182*
@cause
<rdg>, *374*
<rdgGrp>, *377*
<cb>, ***797***
<cell>, *436–439*, ***798***, *813*, *1113*, *1169*
@cert
<corr>, *74*, *348*, *362*
<event>, *414*
<ex>, *345*
<expan>, *345*
<population>, *425*
<trait>, *1195*
<certainty>, *362*, *367*, *621–624*, ***799–801***
<change>, *49*, *51*, *327*, ***802***, *1088*, *1107*
<channel>, *456, 457*, *794*, ***804***, *1069*, *1183*
<char>, *99*, *172*, *178–180*, *246*, ***804***, *805*
<charDecl>, *172*, *177, 178*, *246*, ***805***
<charName>, *172*, *179, 180*, *804, 805*, ***805***
<charProp>, *172*, *177*, *179, 180*, *804*, ***806***, *930*
@children
<iNode>, *588, 589*, *591, 592*, *951*, *1196*
<root>, *588, 589*, *591, 592*, *1112*, *1196*
<choice>, *73–75*, *80*, *88*, *231*, *248*, *343–349*, *360*, *362*, *428*, *489*, *624*, *741*, ***806***, *824*, *887*, *889*, *1044*, *1095*, *1135*
<cit>, *66*, *149*, *157*, *164*, *265–268*, *270*, *272, 273*, *276*, *278, 279*, ***808***, *881*, *970*, *1036, 1037*, *1126*
<cl>, *522–524*, *533*, ***809***
@class
<msItem>, *312*
<msItemStruct>, *1019*
<classCode>, *48*, *51*, ***810***, *1182*
<classDecl>, *51*, *312*, ***810***
<classes>, *635*, *639, 640*, *643*, *654–656*, *666*, *669*, *678*, *812*, ***812***, *875*
<classSpec>, *640*, *655*, *674, 675*, ***812***
<climate>, *424*, ***813***
<closer>, *147–149*, ***815***, *835*, *1066*
<code>, *476*, *628*, ***816***
@code
<occupation>, *1039*
<socecStatus>, *458*, *1139*

<collation>, *315*, ***817***
<collection>, *303, 304*, ***817***
<colloc>, *263*, ***818***
<colophon>, ***819***
@cols
<table>, *436, 437*, *1169*
@columns
<layout>, *294*, *317*, *978, 979*, *1060*
@commodity
<measure>, *85*, *726*, *1001*
<cond>, *566, 567*, ***820***, *914*, *1185*
<condition>, *317*, ***821***
<constitution>, *456, 457*, *794*, ***823***, *1069*, *1183*
@contemporary
<binding>, *777, 778*
<content>, *635*, *640*, *643, 644*, *651*, *666–669*, *671*, *674*, *678*, ***823***, *875*, *999*
@copyOf
<date>, *503*, *512*
<div2>, *512*
<eTree>, *595*, *597*, *599*, *908*
<fs>, *547*
<l>, *504*
<seg>, *504*
<corr>, *73, 74*, *248*, *346–350*, *360*, *362*, *806*, ***824***, *1135*
<correction>, *35*, *51*, *349*, *463, 464*, ***825***, *1176*
@corresp
<anchor>, *492*, *610*
<eLeaf>, *599*, *908*
<eTree>, *599*, *908*
<seg>, *491*
<stage>, *221*
<country>, *83*, *303, 304*, *401*, *403*, *419–421*, *817*, ***826***, *995*, *1057*, *1061*
<creation>, *46*, *51*, ***827***
@cRef
<ref>, *484*
<cRefPattern>, *109, 110*, *483, 484*, ***785***, *1093*, *1176*
<custEvent>, *327*, ***828***, *829*
<custodialHist>, *327*, *748*, *752*, ***829***

<damage>, *363, 364*, ***830***
<damageSpan>, *363, 364*, ***831***
<datatype>, *408*, *636–638*, *653*, *656*, *675*, *766*, ***832***, *875*
<date>, *25*, *27*, *31*, *46*, *51*, *86, 87*, *113, 114*, *116–119*, *121–124*, *147*, *149*, *203–205*, *228, 229*, *282–284*, *294*, *314*, *324–326*, *328*, *412*, *430–433*, *458, 459*, *503*, *512*, *707*, *709, 710*, *752*, *758*, *771, 772*, *774*, *779*, *782*, *815*, *827, 828*, ***832***, ***834***, *835*, *865*, *870, 871*, *956*, *989*, *1011*, *1041*, *1054, 1055*, *1066*, *1075–1077*, *1089, 1090*, *1097*, *1130*, *1167*, *1176*, *1190*, *1235, 1236*
<dateline>, *147–149*, *710*, *784*, *815*, ***835***, *1040, 1041*, *1066*
@datum
<geoDecl>, *926*
<death>, ***836***, *1057*
@decls
<div1>, *463*
<decoDesc>, *294*, *298*, *319, 320*, ***836***, *837*, *1060*
<decoNote>, *320*, *322*, ***837***
<def>, *257–259*, *264, 265*, *267–270*, *275, 276*, *278*, *282–284*, *287–290*, *429*, ***838***, *845*, *879, 880*, *948*, *1085, 1086*, *1126*, *1160*
<default>, *554*, ***839***
@default
<correction>, *463, 464*
<editorialDecl>, *464*
<normalization>, *464*
<defaultVal>, *638*, *677*, ***839***
@defective
<explicit>, *309*, *311*
<incipit>, *309*, *311*, *958*
<msItem>, *309*, *311, 312*
<msItemStruct>, *1019*
@degree
<certainty>, *362*, *367*, *621–624*, *799–801*
<node>, *574, 575*, *935*
<purpose>, *456, 457*, *794*, *1069*, *1078*, *1183*
<del>, *77, 78*, *240*, *247*, *352–357*, *366, 367*, ***840***, *1107*, *1159*
@delim
<refState>, *42*, *111*, *1093*
<delSpan>, *354, 355*, ***841***
<depth>, ***842***, *1002*
<derivation>, *456, 457*, *794*, ***843***, *1069*, *1183*
<desc>, *69, 70*, *76*, *213*, *231*, *233–235*, *241*, *246*, *335–337*, *367*, *408*, *410*, *412*, *416*, *423–425*, *506*, *508*, *621*, *624*, *630*, *633*, *638*, *640*, *653*, *656*, *667*, *675*, *677, 678*, *765, 766*, *812*, ***844***, *875*, *885*, *957*, *962*, *965*, *968*, *991*, *1010*, *1042*, *1063*, *1180*, *1195*, *1210*, *1214*
<dictScrap>, *290*, ***845***
<dimensions>, *294*, *298*, *315*, ***846***, *1060*
@discrete
<sound>, *1140*

<distinct>, *62, 63*, ***848***
<distributor>, *27*, *50, 51*, ***849***, *1176*
<district>, *403*, *419, 420*, ***849, 850***, *995*
, *7*, *66*, *84*, *96*, *100*, *105*, *138*, *140*, *143*, *145–149*, *152*, *157*, *160–162*, *166, 167*, *186*, *190*, *193, 194*, *200*, *209*, *227*, *246*, *252*, *340*, *354*, *384, 385*, *476–478*, *492–494*, *496*, *498, 499*, *510–512*, *514*, *523*, *632*, *657*, *718*, *739*, *747*, *815*, *835*, ***851***, *861*, *903*, *912, 913*, *985*, *1003*, *1066*, *1200*, *1213*
<div1>, *89*, *103, 104*, *107*, *131, 132*, *138, 139*, *141*, *143*, *145*, *152*, *162*, *202*, *208*, *213*, *219, 220*, *446*, *463*, *489, 490*, *508*, *512*, *770*, ***852***, *854*, *861*, *942*, *987*
<div2>, *103, 104*, *130–132*, *138, 139*, *141*, *143*, *208*, *219, 220*, *489, 490*, *512*, *852*, ***854***, *855*, *942*
<div3>, *103, 104*, *629*, *854*, ***855***, *857*
<div4>, *854*, ***857***
<divGen>, *100*, *160*, ***861***
<docAuthor>, *164*, *784*, ***862***
<docDate>, *39*, *164*, ***863***
<docEdition>, ***864***, *1190*
<docImprint>, *39*, *151*, *164*, *863*, ***865***, *1190*
<docTitle>, *39*, *151, 152*, *164*, *200*, *862*, ***866***, *979*, *1181*, *1190, 1191*
<domain>, *456, 457*, *794*, ***868***, *1069*, *1183*
@domains
<joinGrp>, *507*, *966*
<linkGrp>, *474*, *493*, *500–502*
@dur
<date>, *430*, *1236*
<kinesic>, *968*
<pause>, *234*, *248*, *1051*
<recording>, *229, 230*, *1089, 1090*
<time>, *430*, *1235, 1236*
<vocal>, *1214*
@dur-iso
<date>, *709*, *1235*
<time>, *1235*

@ed
<lb>, *979*
<milestone>, *42*, *108*, *1009*
<pb>, *1052*
<refState>, *42*, *111*
<edition>, *25*, *43*, *51*, *119*, *772*, *774*, ***870***, *871*, *1121*
<editionStmt>, *21*, *25*, *51*, *772*, ***871***
<editor>, *51*, *113, 114*, *116*, *122, 123*, *802*, ***872***
<editorialDecl>, *35*, *51*, *463, 464*, ***872***, *1028*, *1176*
<education>, *458*, ***874***
<eg>, ***874***, *888*
<egXML>, ***874***, *937*
<eLeaf>, *593–595*, *597*, *599*, *601, 602*, ***868***, *869*, *908*, *1198*
<elementSpec>, *69, 70*, *213*, *408*, *631*, *633*, *643*, *650, 651*, *653–656*, *666–669*, *671–673*, *678*, ***875***, *885*, *1148*
<email>, *81*, ***877***
<emph>, *8, 9*, *61*, *71*, *151*, *231*, *248*, *505*, *528*, *607*, *611*, *613*, *623, 624*, ***878***
<encodingDesc>, *18*, *33*, *51*, *109*, *460*, *464*, *558, 559*, ***878***, *1176*
@end
<u>, *242–244*
@enjamb
<l>, *189*
<entry>, *252–255*, *257–259*, *262*, *264–270*, *273, 274*, *276*, *278*, *282–284*, *290*, *789*, *808*, *818*, *838*, *845*, ***879***, *886*, *923*, *932*, *934*, *948, 949*, *973*, *981*, *1012*, *1033*, *1036, 1037*, *1053*, *1065*, *1072*, *1085, 1086*, *1158*, *1160*, *1193*, *1225*
<entryFree>, *289*, *797*, ***880***
@eol
<hyphenation>, *51*, *950*
<epigraph>, *66*, *149*, *152*, *164*, *166*, ***881***, *912*
<epilogue>, *201, 202*, ***882***
<equipment>, *229*, ***883, 884***, *1089, 1090*
<equiv>, *69*, *673*, *675*, *766*, *875*, ***885***, *1210*
<eTree>, *593–595*, *597*, *599*, *601, 602*, ***869***, *908*
<etym>, *268, 269*, *276*, *282–284*, *289, 290*, *427*, *845*, *880*, ***886***, *973*, *990*, *1225*
@evaluate
<joinGrp>, *508*
<link>, *475*
<event>, *412*, *414*, *423*, *425*, ***887***
@evidence
<origin>, *1046*
<ex>, *294*, *309*, *311, 312*, *321*, *323*, *340*, *344, 345*, *389*, ***887***
@exclude
<arc>, *586*
<date>, *512*
<div2>, *512*
<eTree>, *595*, *597*
<name>, *512*
<seg>, *510, 511*, *513*

<u>, *510, 511*
<exemplum>, ***888***
<expan>, *88*, *309*, *343–345*, *624*, *741*, *887*, ***889***, *1018, 1019*, *1115*
<explicit>, *299*, *309*, *311, 312*, ***890***, *997*, *1018*
<extent>, *21*, *26*, *113*, *119*, *121*, *124*, *294*, *315*, *772*, ***891***, *1060*, *1097*
@extent
<damageSpan>, *363*
<gap>, *76*, *143*, *240*, *246*, *248*, *311*, *358*, *364*, *922*, *1018*
<orth>, *282, 283*, *286*, *288*
<pron>, *270*, *279*, *282, 283*, *288*, *1072*, *1085*

<f>, *431*, *538–555*, *564*, *566, 567*, *775, 776*, *820*, *839*, ***892***, *894*, *914*, *954*, *1157*, *1168*, *1185*, *1204*, *1206*, *1208*
@facs
<figure>, *340*
<head>, *340*
<locus>, *298*, *997*
<p>, *340*
<pb>, *332*, *340*, *1052*
<w>, *340*
<facsimile>, *333*, *335–337*, *339, 340*, ***895***, *1164*, *1226*
<factuality>, *456, 457*, *794*, ***896***, *1069*, *1183*
<faith>, ***897***
<fDecl>, *561*, *563–565*, *567*, ***892***, *893*, *915, 916*, *1205*, *1208*
<fDescr>, *563–565*, *567*, *892*, ***893***, *916*, *1205*, *1208*
@feats
<fs>, *543*, *547*, *555*
@feature
<shift>, *237*, *1133*
<figDesc>, *444*, *476*, ***898***, *899*, *936*, *1190*, *1243*
<figure>, *90*, *340*, *443–445*, *476*, *898*, ***899***, *936*, *1190*, *1243*
<fileDesc>, *18, 19*, *21, 22*, *50, 51*, *109*, *460*, *558, 559*, *567*, *737*, ***899***, *1176*
<filiation>, *312*, ***901***
@filter
<equiv>, *69*
<finalRubric>, ***902***, *1019*
<fLib>, *542*, *545*, *555*, ***894***
<floatingText>, *158*, *218*, ***903***
<floruit>, *411*, ***905***
<foliation>, *316*, ***906***
@follow
<iNode>, *592*, *951*
<foreign>, *60*, *71*, *248*, *325*, ***906***
<forename>, *300*, *395–399*, *412*, *427*, *747*, ***907***, *924*, *1024*, *1050*, *1056*, *1111*
<forest>, *599*, ***908***
<form>, *257–259*, *261–266*, *268*, *272*, *274*, *276*, *278, 279*, *282–284*, *286–289*, *426–429*, *789*, *797*, *808*, *818*, *838*, *879*, *886*, ***911***, *923*, *932*, *934*, *948, 949*, *953*, *973*, *981*, *990*, *1012*, *1033*, *1035–1037*, *1048*, *1053*, *1065*, *1072*, *1085, 1086*, *1158*, *1160*, *1168*, *1193*, *1203*, *1225*
@form
<objectDesc>, *294*, *314*, *1037*, *1060*, *1163*
<quotation>, *35*, *51*, *872*
<formula>, *315*, *440, 441*, *443*, *817*, ***911***
@from
<app>, *386, 387*
<arc>, *574*, *576–578*, *580*, *582*, *586*, *763*, *935*
<date>, *87*
<locus>, *298*, *309*, *312*, *317*, *319*, *979*, *1019*
<orgName>, *415*
<residence>, *393*
, *528, 529*, *535*, *1145*
<state>, *412*, *416*, *423*
<trait>, *1195*
<front>, *105*, *136*, *151, 152*, *155, 156*, *162*, *164*, *183*, *200, 201*, *218*, *861*, ***912, 913***, *937*, *1129*, *1181, 1182*
<fs>, *431*, *538–541*, *543–548*, *550, 551*, *553*, *555*, *558, 559*, *564*, *566, 567*, *775*, *820*, *914*, ***914***, *954*, *1185*, *1204*, *1206*
<fsConstraints>, *561*, *567*, ***914***, *915*
<fsdDecl>, *558, 559*, *567*, ***917***
<fsDecl>, *558, 559*, *561*, *567*, ***915***, *916, 917*
<fsDescr>, *561*, *567*, *915*, ***916***
<fsdLink>, *559*, *917*, ***917***
@full
<forename>, *396*, *399*
<genName>, *399*
<roleName>, *398*
@function
<cl>, *524*, *809*
<phr>, *524*, *610*, *1060*
<w>, *610*
<funder>, *24*, ***918***
@fVal
<f>, *543*, *545–547*
<fvLib>, *543, 544*, *555, 556*, ***919***
<fw>, *340*, *368*, ***920***

<g>, *99*, *177–180*, *246*, *309*, *343, 344*, *376, 377*, *389*, *760*, ***921***, *972*, *1221*
<gap>, *76*, *143*, *240*, *246*, *248*, *294*, *311*, *358*, *364*, *890*, ***922***, *1018*
<gen>, *264–266*, *789*, ***923***, *1085, 1086*
<genName>, *397–399*, ***924, 925***
<geo>, *418*, *422, 423*, ***925***, *995*
<geoDecl>, ***926***
<geogFeat>, *404, 405*, *429*, ***927***, *928*, *1039*
<geogName>, *403–405*, *429*, *927*, ***928***, *1039*, *1062*
<gi>, *35*, *40*, *49*, *51*, *109*, *628, 629*, *633*, *637, 638*, *812*, *874*, *888*, ***928***, *1028*, *1081*, *1094*, *1125*, *1176*, *1210*
@gi
<tagUsage>, *9*, *37*, *39, 40*, *212*, *1024*, *1172*
@given
<certainty>, *621, 622*, *801*
<gloss>, *68, 69*, *71*, *93*, *268, 269*, *276*, *282–284*, *427*, *638*, *673*, *677*, *875*, ***930***, *1179*, *1210, 1211*
<glyph>, *172*, *177, 178*, *921*, ***930***
<glyphName>, *172*, *177*, *930*, ***931***
@gradual
<writing>, *236*
<gram>, *263*, ***932***
<gramGrp>, *257–259*, *261*, *263–265*, *276*, *282–284*, *288, 289*, *723, 724*, *789*, *818*, *838*, *879*, *923*, *932*, ***934***, *948*, *1012*, *1033*, *1053*, *1065*, *1072*, *1085, 1086*, *1158*, *1160*, *1193*
<graph>, *574–578*, *580*, *582*, *586*, ***935***
<graphic>, *102*, *177*, *333*, *335–337*, *339, 340*, *443–445*, *476*, *895*, *898, 899*, *930*, ***936***, *1164*, *1226*, *1243*
<group>, *136*, *151, 152*, *155, 156*, ***937***, *1182*
@group
<damage>, *364*, *830*
<damageSpan>, *364*

@hand
<add>, *351–353*, *360*
<addSpan>, *354*, *747*
<del>, *352, 353*, *367*, *1159*
<gap>, *358*
<rdg>, *374*, *378*
<restore>, *357*, *1107*
<handDesc>, *294*, *318, 319*, ***938***, *1060*
<handNote>, *319*, *351, 352*, *354*, *360*, *747*, *938*, ***939***, *940*
<handNotes>, *360*, ***940***
@hands
<handDesc>, *318, 319*, *938*
<handShift>, *360*, ***941***
<head>, *57*, *66, 67*, *92, 93*, *100*, *102*, *105*, *113, 114*, *130–132*, *139*, *143*, *145–149*, *151, 152*, *155*, *157, 158*, *161, 162*, *166, 167*, *183, 184*, *200–203*, *206–209*, *252*, *306*, *340*, *382*, *384, 385*, *436–439*, *444*, *476, 477*, *489, 490*, *496*, *498*, *506*, *517*, *629*, *632*, *717, 718*, *770*, *792*, *813*, *851, 852*, *854, 855*, *857*, *861*, *882*, *898, 899*, *913*, *936*, ***942***, *963*, *970*, *987*, *989*, *991*, *1003*, *1129*, *1169*, *1190*, *1200*, *1243*
<headItem>, *94*, ***943***, *944*, *970*
<headLabel>, *94*, *943*, ***944***, *970*
<height>, *294*, *298*, *315*, *846*, ***945***, *1002*, *1060*
@height
<graphic>, *1243*
<heraldry>, *302*, ***946***
<hi>, *37*, *39*, *62*, *71*, *158*, *184*, *190*, *223*, *345*, *367*, *401*, *489*, *717, 718*, *770*, *797*, ***947***, *1032*, *1191*, *1213*
<history>, *293, 294*, *324, 325*, ***947***
<hom>, *255*, *257*, *276*, ***948***
<hyph>, *257–259*, *261*, *289*, *838*, ***949***, *1160*, *1168*
<hyphenation>, *51*, ***950***

<ident>, *35*, *628*, ***953***
@ident
<application>, *44*, *761, 762*
<attDef>, *70*, *213*, *408*, *638, 639*, *653*, *655, 656*, *666*, *675*, *677*, *765, 766*, *875*, *885*
<classSpec>, *640*, *655*, *674, 675*, *812*
<elementSpec>, *69, 70*, *213*, *408*, *631*, *633*, *643*, *650, 651*, *653–656*, *666–669*, *671–673*, *678*, *875*, *885*, *1148*
<language>, *47*, *51*, *874*, *975, 976*, *1069*
<macroSpec>, *674*, *999*
<moduleSpec>, *630*, *1010*
<schemaSpec>, *3–5*, *213*, *642–644*, *650*, *657*, *672*, *1120*
<valItem>, *69*, *213*, *408*, *638*, *653*, *675*, *677*, *885*, *1210, 1211*
<idno>, *27, 28*, *121*, *292–294*, *303–305*, *328, 329*, *350*, *757*, *817*, ***954***, *959*, *1016, 1017*, *1020*, *1102*, *1121*, *1128*, *1167*, *1176*
<if>, *564*, *567*, ***954***
<iff>, *566, 567*, *775*, ***954***
<imprimatur>, *164*, ***956***
<imprint>, *31*, *51*, *113*, *116, 117*, *119*, *121–124*, *758*, *774*, ***956***, *989*, *1011*, *1077*, *1097*
<incident>, *231*, *233, 234*, ***957***

<incipit>, *294*, *299*, *309*, *311, 312*, ***958***, *997*, *1018*
@inDegree
<node>, *576–578*, *580*, *582*, *1027*
<index>, *81*, *98–100*, ***958***, *1178*
@indexName
<index>, *99*, *958*, *1178*
<iNode>, *588, 589*, *591, 592*, ***951***, *1196*
<institution>, *305*, ***959***, *1102*
<interaction>, *456, 457*, *794*, ***961***, *1069*, *1183*
<interp>, *530*, *533, 534*, ***961***, *962*
<interpGrp>, *530*, *533, 534*, ***962***
<interpretation>, *35*, *51*, ***962***
@interval
<when>, *241–243*, *501*, *1187*, *1217*
<item>, *7*, *47, 48*, *51*, *65*, *84, 85*, *90–94*, *96*, *109*, *114*, *148*, *161, 162*, *166*, *212*, *315*, *320, 321*, *349*, *458*, *477*, *506*, *625*, *770*, *810*, *942–944*, ***963***, *965*, *967*, *970, 971*, *987, 988*, *1003*, *1081*, *1106*, *1129*, *1146*, *1176*, *1182*, *1199*
@iterated
<kinesic>, *241*, *968*
<vocal>, *1214*
<iType>, *262*, ***953***

<join>, *220*, *506–508*, *530*, *613*, ***965***, *966*
<joinGrp>, *507, 508*, ***966***

@key
<country>, *419*, *826*, *995*, *1057*
<geogName>, *403, 404*, *429*
<memberOf>, *635*, *639, 640*, *643*, *654–656*, *666*, *669*, *678*, *812*, *875*, *1004*
<moduleRef>, *3–5*, *213*, *642–644*, *650*, *657*, *1010*, *1120*
<name>, *80*, *294*, *392*, *395*, *400*, *402*, *404*, *412*
<nationality>, *410*, *1025*
<orgName>, *400*, *1043*
<persName>, *395*, *397–399*
<placeName>, *402, 403*, *405*, *423*, *1039*, *1043*
<rs>, *80*, *394*, *400*, *402*
<socecStatus>, *409*
<specDesc>, *629*, *1147, 1148*, *1150*
<trait>, *410*
<keywords>, *47, 48*, *51*, *810*, ***967***, *1182*
<kinesic>, *231*, *241*, ***968***

<l>, *7*, *61*, *66*, *77, 78*, *95*, *103, 104*, *108*, *128–132*, *152*, *157, 158*, *183–196*, *201–203*, *210, 211*, *215–220*, *354–356*, *360*, *364*, *374, 375*, *378, 379*, *384–388*, *469*, *471–473*, *504*, *506*, *514*, *517*, *523*, *606–609*, *611–613*, *615*, *726*, *782*, *786*, *830*, *840*, *881, 882*, *903*, *941*, *965*, ***969***, *979*, *984*, *1044*, *1071*, *1109*, *1181*, *1223*
<label>, *44*, *84*, *91–94*, *109*, *149*, *162*, *212*, *349*, *410*, *412*, *424*, *574–578*, *580*, *582*, *586*, *588, 589*, *591–595*, *597*, *599*, *601, 602*, *761–763*, *868, 869*, *887*, *908*, *935*, *943, 944*, *951*, ***970, 971***, *982*, *1027*, *1066*, *1081*, *1112*, *1129*, *1154*, *1195, 1196*, *1198*
@label
<rhyme>, *195*, *1109*
<lacunaEnd>, *383*, ***971***
<lacunaStart>, ***972***
<lang>, *268, 269*, *276*, *282–284*, *289, 290*, *427*, *845*, *880*, *886*, ***973***
@lang
<code>, *816*
<langKnowledge>, *409*, *458*, ***974***
<langKnown>, *409*, *458*, *974*, ***975***
<language>, *47*, *51*, *874*, *975*, ***976***, *1069*
<langUsage>, *47*, *51*, *874*, ***975***, *976*, *1069*
<layout>, *294*, *317*, ***978***, *979*, *1037*, *1060*
<layoutDesc>, *294*, *317*, ***979***, *1037*, *1060*
<lb>, *39*, *128*, *149*, *164*, *297*, *299*, *311*, *321*, *340*, *608*, *717, 718*, ***979***, *997*, *1019*, *1153*
<lbl>, *262*, *274–276*, *282, 283*, ***981***
<leaf>, *588, 589*, *591, 592*, ***982***, *1112*, *1196*
<lem>, *373*, *376–378*, *383*, *385–388*, *760*, *972*, ***983***, *1084*, *1220, 1221*
@lemma
<w>, *526*, *1215*
@lemmaRef
<w>, *1215*
@length
<refState>, *42*, *1093*
@level
<langKnown>, *409*, *458*, *974, 975*
<title>, *28*, *114*, *116–119*, *121–123*, *125*, *326*, *758*, *771*, *1011*, *1127*
<lg>, *108*, *128–130*, *152*, *184–186*, *190, 191*, *193–196*, *201*, *216–220*, *379*, *607*, *609*, *611*, *613*, *882*, ***984***, *1109*
<link>, *196*, *243, 244*, *446*, *469*, *472–475*, *491*, *493*, *497, 498*, *500–502*, *505*, *512*, *531*, *535*, *556, 557*, *985*, ***985***
<linkGrp>, *196*, *243, 244*, *446*, *473, 474*, *491*, *493*, *497, 498*, *500–502*, *531*, *535*, *556, 557*, *985*, ***985***

<list>, *7*, *47, 48*, *51*, *65*, *84, 85*, *90–94*, *96*, *109*, *114*, *146*, *148*, *161, 162*, *166*, *212*, *315*, *320, 321*, *349*, *458*, *477*, *506*, *625*, *770*, *810*, *942–944*, *963*, *965*, *967*, *970, 971*, ***987, 988***, *1003*, *1081*, *1106*, *1129*, *1146*, *1176*, *1182*, *1199*
<listBibl>, *100*, *113*, *326*, *381*, *718*, *748*, *861*, ***989***
<listNym>, *426*, ***990***
<listOrg>, *416*, ***991***
<listPerson>, *231*, *234*, *240*, *246*, *407*, ***992***, *1050*, *1099*, *1133*
<listPlace>, *421, 422*, *425*, ***992***
<listRef>, *633*, ***993***
<listWit>, *380–382*, ***993***, *1222*
@loc
<app>, *379*, *384*
<locale>, *459*, ***995***, *1130*
<localName>, *172*, *177*, *179, 180*, *804*, *806*, *930*, ***994***
<location>, *418–420*, *422, 423*, ***995, 996***
@location
<pos>, *288*
<variantEncoding>, *384–387*, *1212*
<locus>, *298, 299*, *307*, *309*, *311, 312*, *315–317*, *319–321*, *979*, ***997***, *1015*, *1018, 1019*, *1115*
@locus
<certainty>, *362*, *367*, *621–624*, *799–801*
<respons>, *362*, *625*, *1106*
@lrx
<surface>, *335–337*, *339*, *895*, *1164*, *1226*
<zone>, *335–337*, *339*, *1226*
@lry
<surface>, *335–337*, *339*, *895*, *1164*, *1226*
<zone>, *335–337*, *339*, *1226*

<m>, *526, 527*, *533*, ***998***, *1215*
<macroSpec>, *674*, ***999***
@mainLang
<textLang>, *294*, *312, 313*, *1018, 1019*, *1184*
<mapping>, *172*, *178–180*, *804, 805*, ***999***
@marks
<quotation>, *35*, *51*, *872*, *1081*
@matchPattern
<cRefPattern>, *109, 110*, *483, 484*, *785*, *1093*, *1176*
<material>, *293*, *296*, *314*, *322*, *1037*, *1163, 1164*, *1216*
@material
<supportDesc>, *294*, *1037*, *1060*
@max
<numeric>, *541*, *550*, *1034*
@maxOccurs
<datatype>, *653*, *832*
<measure>, *84, 85*, *315*, *405*, *419*, *726*, ***1001***, *1062*
<measureGrp>, *85*, ***1002***
@medium
<handNote>, *360*, *938*, *940*
<handShift>, *360*, *941*
<meeting>, *120*, ***1003***
<memberOf>, *635*, *639, 640*, *643*, *654–656*, *666*, *669*, *678*, *812*, *875*, ***1004***
<mentioned>, *60*, *67, 68*, *95*, *157*, *268–270*, *276*, *289*, *375*, *378*, *427*, *526*, *620*, *886*, *973*, *990*, ***1005***, *1030*, *1225*, *1243*
@mergedIn
<orth>, *287*, *1085, 1086*
<pron>, *287*
@met
, *190*, *193, 194*
<l>, *191, 192*, *969*
<lg>, *191*, *194*
<seg>, *193*
<metDecl>, *193*, *197*, ***1006***
@method
<correction>, *51*
<normalization>, *1028*
<variantEncoding>, *384–387*, *1212*
<metSym>, *193*, *197*, *1006*, ***1007***
<milestone>, *42*, *107, 108*, ***1009***
@mimeType
<binaryObject>, *777*
<equiv>, *69*
<graphic>, *102*
@minOccurs
<datatype>, *832*
@mode
<alt>, *513, 514*, *755*
<altGrp>, *514*, *756*
<attDef>, *70*, *213*, *408*, *653*, *655, 656*, *666*
<channel>, *456, 457*, *794*, *804*, *1069*, *1183*
<classSpec>, *655*
<classes>, *654, 655*
<elementSpec>, *70*, *213*, *408*, *643*, *650, 651*, *653–656*, *667*, *885*
<memberOf>, *654*
@module
<classSpec>, *674, 675*, *812*
<elementSpec>, *69, 70*, *408*, *633*, *650, 651*, *653–655*, *669*, *673*, *678*, *875*
<macroSpec>, *674*, *999*

<moduleRef>, *3–5*, *213*, *642–644*, *650*, *657*, ***1010***, *1120*
<moduleSpec>, *630*, ***1010***
<monogr>, *31*, *51*, *113*, *116, 117*, *119–124*, *758*, *774*, *989*, ***1011***, *1097*
<mood>, ***1012***, *1053*, *1193*
<move>, *214*, ***1014***
<msContents>, *293, 294*, *307*, *309*, *311, 312*, *901*, ***1015***, *1016*
<msDesc>, *292–294*, *329*, *350*, *901*, ***1016***, *1020*
<msIdentifier>, *292–294*, *303–305*, *329*, *350*, *817*, *959*, *1016*, ***1017***, *1020*, *1102*
<msItem>, *294*, *298, 299*, *307*, *309*, *311, 312*, *901*, *997*, *1015, 1016*, ***1018***
<msItemStruct>, ***1019***
<msName>, *303–305*, *329*, *817*, *1020*, ***1020***
<msPart>, *329*, ***1020***
<musicNotation>, *321*, ***1022***
@mutual
<relation>, *413, 414*, *992*, *1050*, *1099*
@my:topic
<p>, *657*

@n
<ab>, *489*
<addSpan>, *354*, *747*
<app>, *377*
<bibl>, *718*
<body>, *105*
<cb>, *797*
<change>, *49*, *802*
<div1>, *103, 104*, *107*, *131, 132*, *138, 139*, *141*, *145*, *152*, *208*, *213*, *219, 220*, *463*, *489, 490*, *852*, *854*, *942*
<div2>, *103, 104*, *130–132*, *138, 139*, *141*, *208*, *219, 220*, *489, 490*, *852*, *854, 855*
<div3>, *103, 104*, *854, 855*, *857*
<div4>, *854*, *857*
, *7*, *84*, *105*, *138*, *140*, *143*, *146–149*, *157*, *186*, *190*, *209*, *246*, *340*, *384, 385*, *476, 477*, *739*, *851*
<divGen>, *100*, *861*
<eTree>, *593–595*, *597*, *599*, *869*, *908*
<edition>, *25*, *870, 871*
<entry>, *253*, *255*, *259*, *266*, *274*, *1160*
<fLib>, *542*, *545*, *894*
<figure>, *445*
<forest>, *599*, *908*
<formula>, *443*
<front>, *105*
<fvLib>, *543, 544*, *555, 556*, *919*
<head>, *105*
<hom>, *255*
<item>, *7*, *92*, *161, 162*, *477*, *770*, *963*, *987*
<l>, *7*, *103, 104*, *191–194*, *375*, *384–388*
<lb>, *608*
<lg>, *128*, *130*, *152*, *186*, *190*, *193*
<metSym>, *197*
<milestone>, *42*, *107, 108*, *1009*
<msItem>, *298, 299*, *307*, *309*, *312*, *997*, *1015*
<msItemStruct>, *1019*
<name>, *751*
<note>, *95*, *1030*
<p>, *105*, *657*
<pb>, *158*, *363*, *1052*
<region>, *1096*
<s>, *554*, *557*, *985*
<said>, *611*
<seal>, *323*, *1122*
<seg>, *192*, *489*, *612*
<sense>, *254, 255*, *257–259*, *264–266*, *270*, *272*, *275, 276*, *282–284*, *287, 288*, *879*, *1085, 1086*, *1126*, *1160*
<text>, *105*
<textDesc>, *456, 457*, *794*, *1069*, *1183*
<titlePage>, *105*
<tree>, *588, 589*, *591, 592*, *1196*
<name>, *9*, *23–25*, *28*, *50, 51*, *61*, *73*, *79–82*, *116*, *119, 120*, *129*, *132*, *147*, *164*, *166, 167*, *201*, *203–205*, *223*, *229*, *294*, *299, 300*, *310*, *323–325*, *327*, *345*, *348*, *360*, *392*, *394, 395*, *400–402*, *404, 405*, *412*, *414*, *416*, *428, 429*, *438*, *458, 459*, *491*, *508*, *512*, *567*, *620*, *718*, *743*, *751*, *753*, *779*, *782*, *802*, *815*, *821*, *835, 836*, *865*, *871*, *874*, *882*, *928*, *997*, ***1023***, *1039*, *1041, 1042*, *1046*, *1054*, *1069*, *1088–1090*, *1104*, *1122*, *1127, 1128*, *1130*, *1137*, *1156*, *1176*, *1190*, *1192*, *1213*
@name
<attRef>, *766*
<equiv>, *69*, *885*, *1210*
<f>, *431*, *538–555*, *564*, *566, 567*, *775, 776*, *820*, *839*, *892*, *894*, *914*, *954*, *1157*, *1168*, *1185*, *1204*, *1206*, *1208*
<fDecl>, *561*, *563–565*, *567*, *892, 893*, *915, 916*, *1205*, *1208*
<namespace>, *37*, *1024*, *1172*

<relation>, *413, 414*, *425, 426*, *992*, *1050*, *1098, 1099*
<vLabel>, *546*, *1206*
<nameLink>, *398*, ***1024***
<namespace>, *37*, ***1024***, *1172*
<nationality>, *410*, ***1025***
@new
<handShift>, *360*
<shift>, *233*, *237*, *1133*
@next
<cl>, *523*
<def>, *288*
<lg>, *219*
<oVar>, *279*
<orth>, *287*
<pron>, *287*
<q>, *508*
<seg>, *612*
<node>, *574–578*, *580*, *582*, *586*, *935*, ***1027***
@norm
<orth>, *286*
<pos>, *724*
<tns>, *286*
<usg>, *286*
<normalization>, *35*, *51*, *463, 464*, *872*, ***1028***, *1176*
@notAfter
<acquisition>, *325*
<affiliation>, *753*
<age>, *754*
<application>, *44*, *762*
<birth>, *393*
<custEvent>, *327*, *829*
<date>, *87*
<death>, *1057*
<education>, *874*
<event>, *425*
<floruit>, *411*, *905*
<orgName>, *415, 416*
<origDate>, *294*, *324*, *1045*
<origin>, *325*, *1046*
<persName>, *413*
<placeName>, *418, 419*, *422*
<population>, *425*
<residence>, *393*, *1050*, *1103*
<state>, *423*
@notation
<formula>, *440, 441*, *911*
@notBefore
<acquisition>, *325*
<affiliation>, *753*
<birth>, *300*, *393*
<custEvent>, *327*, *829*
<date>, *87*
<death>, *1057*
<education>, *874*
<event>, *425*
<floruit>, *411*, *905*
<nationality>, *410*, *1025*
<orgName>, *416*
<origDate>, *294*, *324*, *1045*
<origin>, *325*, *1046*
<persName>, *413*
<placeName>, *418, 419*, *422*
<population>, *425*
<region>, *422*
<residence>, *1103*
<state>, *416*, *423*
<trait>, *410*
<note>, *29*, *49*, *51*, *95*, *113, 114*, *119*, *121*, *125*, *203*, *229*, *246*, *268*, *276*, *305*, *307*, *328*, *345*, *347, 348*, *375*, *389*, *469*, *471–473*, *620*, *813*, *1015*, ***1030***, *1031*, *1089*
<notesStmt>, *21*, *29*, *51*, ***1031***
@ns
<attDef>, *653*, *656*
<elementSpec>, *69*, *654–656*
<schemaSpec>, *657*
<num>, *83*, *438*, *1001*, ***1032***
<number>, *264*, *279*, *282–284*, *288*, *789*, *1012*, ***1033***, *1053*, *1193*
<numeric>, *431*, *540, 541*, *550*, *552*, ***1034***
<nym>, *426–429*, *990*, ***1035***
@nymRef
<forename>, *427*
<geogFeat>, *429*
<name>, *428*

<objectDesc>, *294*, *314*, *1016*, ***1037***, *1060*, *1163*
<occupation>, *300*, *427*, *458*, ***1039***, *1050*
@occurs
<tagUsage>, *39, 40*, *1024*, *1172*
<offset>, *405*, *419*, *424*, *430, 431*, *996*, ***1039***, *1062*
<opener>, *147–149*, *710*, ***1040, 1041***, *1066*
@opt
<def>, *288*
<orth>, *287*

<pron>, *287*
@ord
<iNode>, *589*, *1196*
<root>, *589*, *1196*
<tree>, *588, 589*, *591, 592*, *1196*
@order
<graph>, *574–578*, *580*, *582*, *586*, *935*
<tree>, *588, 589*, *591, 592*, *1196*
<oRef>, *266*, *270*, *278, 279*, ***1036***
<org>, *415, 416*, *991*, ***1042***
@org
<attList>, *639*
<div1>, *143*
, *227*
<vColl>, *547–549*, *551*, *553*, *1204*, *1207*
<vMerge>, *553*, *1207*
<orgName>, *30*, *400, 401*, *415, 416*, *753*, *991*, *1042*, ***1043***, *1089*
<orig>, *75*, *80*, *231*, *806*, ***1044***, *1095*
@orig
<orth>, *286, 287*, *1085, 1086*
<pron>, *287*
<tns>, *286*
<usg>, *286*, *1085, 1086*
<origDate>, *293, 294*, *324*, ***1045***
<origin>, *294*, *324, 325*, *947*, ***1046***
@origin
<timeline>, *241–245*, *501*, *1187*
<origPlace>, *293, 294*, *324*, ***1045***
<orth>, *257–259*, *261–266*, *268*, *272*, *274*, *276*, *278, 279*, *282–284*, *286–290*, *427*, *789*, *808*, *818*, *838*, *845*, *879, 880*, *886*, *911*, *923*, *932*, *934*, *948, 949*, *953*, *973*, *981*, *1012*, *1033*, *1035–1037*, ***1048***, *1053*, *1065*, *1072*, *1085, 1086*, *1158*, *1160*, *1168*, *1193*, *1203*, *1225*
@otherLangs
<textLang>, *313*, *1184*
@outDegree
<iNode>, *951*
<node>, *576–578*, *580*, *582*, *1027*
<root>, *1112*
<oVar>, *279*, *789*, *1012*, ***1037***, *1053*, *1193*

<p>, *8, 9*, *27*, *30*, *33–35*, *37*, *40*, *51*, *57*, *66*, *75*, *77–81*, *84*, *87, 88*, *98, 99*, *102*, *105*, *107*, *109*, *111, 112*, *130*, *132, 133*, *139–143*, *145–149*, *151, 152*, *157, 158*, *161, 162*, *166, 167*, *177*, *197*, *200–205*, *208*, *210*, *213, 214*, *217–221*, *223, 224*, *228, 229*, *246*, *292–294*, *296–298*, *302*, *307*, *314–328*, *340*, *349*, *363*, *409*, *412*, *414*, *416*, *423*, *441*, *443, 444*, *457–459*, *463, 464*, *476–478*, *481*, *483, 484*, *489*, *492, 493*, *496*, *503*, *506*, *508*, *522*, *524*, *528–530*, *567*, *606–608*, *613*, *615*, *628, 629*, *631–633*, *638*, *657*, *737*, *749*, *752*, *764*, *769, 770*, *772*, *777, 778*, *788*, *806*, *812*, *815*, *817*, *821*, *825*, *831*, *835–837*, *841*, *851, 852*, *854, 855*, *857*, *861*, *871, 872*, *874*, *878*, *883, 884*, *888*, *899*, *901*, *903*, *912, 913*, *928*, *938, 939*, *942*, *946, 947*, *950*, *962*, *965*, *974*, *978, 979*, *985*, *988*, *1003*, *1014–1016*, *1022*, *1028*, *1032*, *1037*, *1041*, ***1048***, *1050*, *1052*, *1054, 1055*, *1057*, *1060*, *1066*, *1070*, *1076*, *1081*, *1088–1090*, *1094*, *1099, 1100*, *1117*, *1119*, *1122*, *1125*, *1129, 1130*, *1140*, *1142, 1143*, *1145, 1146*, *1149*, *1155*, *1167*, *1176*, *1190*, *1200*, *1213*, *1216*
@parent
<iNode>, *588, 589*, *591, 592*, *951*, *1196*
<leaf>, *588, 589*, *591, 592*, *982*, *1196*
@part
<cl>, *523*
, *143*
<l>, *129–132*, *184*, *202, 203*, *215, 216*, *881*, *969*
<lg>, *130*, *216*, *218–220*
<seg>, *188*, *488*, *613*
<particDesc>, *407*, *457, 458*, ***1050***
@parts
<nym>, *428*
@passive
<interaction>, *456*, *961*, *1183*
<relation>, *413, 414*, *425, 426*, *1098, 1099*
@pattern
<metDecl>, *193*, *197*, *1006*
<pause>, *231*, *234*, *237*, *239*, *246*, *248*, ***1051***, *1133*
<pb>, *90*, *97*, *158*, *162*, *166*, *299*, *332*, *340*, *363*, *797*, *997*, ***1052***
<per>, *1012*, ***1053***, *1193*
@perf
<move>, *1014*
<performance>, *200*, *203–205*, *1014*, ***1054, 1055***
<persName>, *30*, *49*, *300*, *395–399*, *407*, *411–413*, *416*, *419*, *427*, *620*, *622, 623*, *625*, *747*, *799*, *801*, *907*, *924, 925*, *996*, *1003*, *1024*, *1042*, *1046*, *1050*, ***1056***, *1057*, *1104*, *1111*
<person>, *231*, *234*, *240*, *246*, *300*, *407*, *411–413*, *427*, *458*, *887*, *992*, *1050*, ***1057***, *1099*, *1133*, *1154*

<personGrp>, *992*, ***1059***
<phr>, *487*, *524*, *526*, *533*, *554*, *610*, ***1060***
<physDesc>, *293, 294*, *1016*, ***1060***
<place>, *418–426*, *813*, *992*, *995, 996*, ***1061***, *1180*
@place
<add>, *77, 78*, *132*, *352, 353*, *360*, *746*
<fw>, *368*, *920*
<note>, *95*, *119*, *121*, *203*, *375*, *469*, *471–473*
<placeName>, *149*, *395*, *402, 403*, *405*, *411*, *418–426*, *620, 621*, *623*, *710*, *813*, *849, 850*, *992*, *995, 996*, *1039*, *1043*, *1056, 1057*, ***1062***, *1066*, *1089*, *1096*, *1103*, *1131*, *1180*
<population>, *425*, ***1063***
<pos>, *257–259*, *261*, *263–265*, *276*, *282–284*, *288, 289*, *723, 724*, *789*, *838*, *879*, *923*, *934*, *948*, *1033*, ***1065***, *1072*, *1085*, *1160*
<postBox>, ***1065***
<postCode>, *82, 83*, *751*, ***1065, 1066***
<postscript>, *149*, ***1066***
<pRef>, *279*
@prefix
<schemaSpec>, *644*, *672*, *1120*
<preparedness>, *456, 457*, *794*, ***1068***, *1069*, *1183*
@prev
<cl>, *523*
<def>, *288*
<lg>, *219*
<oVar>, *279*
<orth>, *287*
<pron>, *287*
<seg>, *612*
<principal>, *24*, ***1068***
<profileDesc>, *18*, *51*, *407*, ***1069***
<projectDesc>, *33*, ***1070***, *1176*
<prologue>, *201*, ***1071***
<pron>, *257–259*, *261–265*, *270*, *276*, *279*, *282–284*, *287–290*, *838*, *845*, *879, 880*, *948, 949*, *953*, *1033*, ***1072***, *1085*, *1160*
<provenance>, *294*, *324, 325*, *947*, ***1073***
<ptr>, *44*, *88*, *90*, *126*, *161*, *166*, *269*, *274, 275*, *284*, *380*, *443*, *445*, *468, 469*, *475*, *478*, *480, 481*, *484*, *508*, *602*, *629*, *633*, *759*, *761, 762*, *813*, *993*, ***1074***, *1222*
<publicationStmt>, *21, 22*, *27*, *50, 51*, *113*, *567*, *737*, *772*, *899*, *1075*, ***1076***, *1176*
<publisher>, *27*, *31*, *39*, *51*, *113, 114*, *116*, *119*, *121–124*, *164*, *771*, *772*, *774*, *956*, *989*, *1075, 1076*, ***1077***, *1097*, *1121*
<pubPlace>, *27*, *31*, *39*, *51*, *113, 114*, *116*, *119*, *121–124*, *164*, *326*, *772*, *774*, *865*, *956*, ***1075***, *1076, 1077*, *1097*
<purpose>, *456, 457*, *794*, *1069*, ***1078***, *1183*

<q>, *8*, *57*, *60, 61*, *71*, *78–80*, *133*, *149*, *151*, *161*, *167*, *278*, *345*, *412*, *483*, *486*, *503*, *505, 506*, *508*, *708*, *749*, *878*, *912*, *946*, *965*, *971*, *1048*, ***1080***, *1095*, *1114*, *1122*
@quantity
<depth>, *842*, *1002*
<height>, *846*, *945*, *1002*
<measure>, *84, 85*, *315*, *405*, *726*, *1001*
<space>, *367*, *1144*
<width>, *846*, *1002*
<quotation>, *35*, *51*, *872*, ***1081***
<quote>, *66*, *149*, *157, 158*, *164*, *265–270*, *272, 273*, *276*, *278, 279*, *293, 294*, *302*, *321*, *323*, *381*, *412*, *469*, *471–473*, *808*, *881*, *912*, *970*, *1036, 1037*, ***1082***, *1126*

<rdg>, *350*, *357*, *372–374*, *376–379*, *382–389*, *760*, *971, 972*, *983*, ***1083***, *1084*, *1219–1221*
<rdgGrp>, *376–378*, *760*, ***1084***, *1220*
<re>, *276*, ***1085, 1086***
@real
<l>, *191, 192*
<seg>, *192*
@reason
<gap>, *76*, *143*, *248*, *311*, *358*, *364*, *922*, *1018*
<supplied>, *358*, *364*, *367*, *1019*, *1162*
<unclear>, *77*, *248*, *364*, *1200*
<recordHist>, *326*, *748*, *752*, ***1088***
<recording>, *228–230*, ***1089***, *1090*
<recordingStmt>, *228, 229*, ***1090***
<ref>, *66*, *89*, *96, 97*, *123*, *126*, *162*, *166*, *266*, *274, 275*, *282, 283*, *289*, *312*, *314*, *319*, *323*, *326*, *378*, *381*, *445*, *471, 472*, *476–478*, *484*, *718*, *854*, *886*, *901*, *970*, *973*, *1019*, *1082*, *1088*, ***1091***, *1122*, *1141*, *1225*
@ref
<forename>, *427*
<g>, *99*, *177–180*, *246*, *309*, *343, 344*, *376, 377*, *389*, *760*, *921*, *972*, *1221*
<name>, *300*, *392*, *412*, *1023*
<persName>, *49*
<state>, *1154*
<surname>, *395*

<refsDecl>, *40*, *42*, *109–111*, *483, 484*, ***1093, 1094***, *1176*
<refState>, *42*, *111*, ***1093***
<reg>, *75*, *80*, *231*, *806*, *1044*, ***1095***
<region>, *303, 304*, *403*, *405*, *411*, *419*, *422*, *817*, *1062*, ***1096***, *1103*, *1131*
<relatedItem>, *123, 124*, ***1097***
<relation>, *413, 414*, *425, 426*, *992*, *1050*, ***1098***, *1099*, ***1099***
<relationGrp>, *414*, *992*, *1050*, ***1099***
<remarks>, *633*, *638*, *812*, ***1100***
@rend
<castGroup>, *206, 207*, *790*
<date>, *503*
<del>, *77, 78*, *352*, *367*, *840*
<delSpan>, *354, 355*
<emph>, *9*, *61*, *528*
<figure>, *476*
<foreign>, *71*
<formula>, *443*
<gloss>, *68*, *71*
<graph>, *935*
<head>, *132*, *145*, *151*, *207*, *436*, *490*, *717*, *792*
<headItem>, *943, 944*
<headLabel>, *943, 944*
<hi>, *62*, *71*, *345*, *367*, *717*, *770*, *797*, *947*, *1032*
<l>, *128, 129*
<list>, *91, 92*, *971*
<name>, *9*, *61*, *203*
<note>, *203*
<p>, *340*
<pron>, *283*
<ptr>, *469*
<q>, *71*, *503*
<ref>, *471, 472*
<rubric>, *299*, *997*
<said>, *64*, *1117*
<seg>, *1124*
<soCalled>, *528*
<stage>, *131, 132*, *221*, *490*, *1152*
<table>, *436*
<term>, *68*, *90*, *930*, *1179*
<title>, *71*
@render
<tagUsage>, *9*, *37*, *1024*, *1172*
<rendition>, *9*, *37, 38*, *61*, *718*, ***1101***, *1172*
@rendition
<docDate>, *39*
<docImprint>, *39*
<docTitle>, *39*
<emph>, *61*
<head>, *718*
<hi>, *39*, *718*
<p>, *9*, *37*
<pubPlace>, *39*
<publisher>, *39*
<titlePart>, *39*
@replacementPattern
<cRefPattern>, *109, 110*, *483, 484*, *785*, *1093*, *1176*
<repository>, *292–294*, *303–305*, *329*, *350*, *757*, *817*, *959*, *1016, 1017*, *1020*, ***1102***
<residence>, *393*, *411*, *458*, *1050*, ***1103***
<resp>, *23–25*, *28*, *30*, *49–51*, *73*, *116*, *119, 120*, *229*, *310*, *345*, *360*, *567*, *782*, *802*, *871*, *1089, 1090*, *1104*, ***1104***, *1127, 1128*, *1176*, *1192*
@resp
<add>, *360*
<corr>, *73*, *348–350*, *360*, *362*
<delSpan>, *354*
<event>, *414*
<ex>, *345*
<expan>, *345*
<handShift>, *360*
<interpGrp>, *530*, *962*
<note>, *375*, *620*
<origin>, *1046*
<population>, *425*, *1063*
<rdg>, *374*, *389*
<rdgGrp>, *377*
<reg>, *1095*
<respons>, *362*, *625*, *1106*
<spanGrp>, *528, 529*
<supplied>, *358*, *364*, *367*
<witDetail>, *378*, *1220*
<respons>, *362*, *625*, ***1106***
<respStmt>, *23–25*, *28*, *30*, *49–51*, *73*, *116*, *119, 120*, *229*, *310*, *345*, *360*, *567*, *782*, *802*, *871*, *1089, 1090*, *1104*, ***1104***, *1127, 1128*, *1176*, *1192*
<restore>, *357*, ***1107***
@result
<join>, *506*, *613*, *965*
<joinGrp>, *507, 508*, *966*
<revisionDesc>, *18*, *49*, *51*, *460*, *802*, ***1107***, *1176*
<rhyme>, *195, 196*, ***1109***
@rhyme
, *190*, *193*
<lg>, *191*, *194–196*, *1109*

<role>, *203*, *205–207*, *210, 211*, *214*, *744*, *790–792*, *1014*, *1054*, ***1110***
@role
<cell>, *436, 437*, *439*, *798*, *813*, *1113*, *1169*
<editor>, *872*
<person>, *1057*
<personGrp>, *1059*
<row>, *437*, *1113*, *1169*
<roleDesc>, *206, 207*, *744*, *790*, *792*, *1110*, ***1110***
<roleName>, *398, 399*, *747*, *907*, *1056*, ***1111***
<root>, *588, 589*, *591, 592*, ***1112***, *1196*
<row>, *436–439*, *798*, *813*, ***1113***, *1169*
@rows
<table>, *436, 437*, *1169*
<rs>, *46*, *78–80*, *203, 204*, *394*, *400*, *402*, *985*, *1054, 1055*, ***1114***
<rubric>, *294*, *297*, *299*, *890*, *997*, *1019*, ***1115***, *1153*
@ruledLines
<layout>, *317*, *978, 979*, *1037*

<s>, *396*, *488*, *494*, *505*, *512*, *522*, *524*, *526*, *528–530*, *532–534*, *554*, *556, 557*, *759*, *965*, *985*, ***1116***, *1124*
<said>, *64–66*, *412*, *607*, *611*, ***1117***
<salute>, *147–149*, *167*, *710*, *770*, *815*, *1041*, *1066*, ***1118***
@sameAs
<date>, *503*
@sample
, *143*, *227*
<samplingDecl>, *33, 34*, *51*, ***1119***
@scale
<graphic>, *899*
<schemaSpec>, *3–5*, *213*, *642–644*, *650*, *657*, *672*, ***1120***
@scheme
<att>, *764*
<catRef>, *48*
<classCode>, *48*, *51*, *810*, *1182*
<gi>, *928*
<keywords>, *47, 48*, *51*, *810*, *967*, *1182*
<locus>, *299*, *316*
<occupation>, *1039*
<rendition>, *9*, *38*, *61*, *718*, *1101*
<socecStatus>, *458*, *1139*
@scope
<dimensions>, *294*, *1060*
<handNote>, *319*, *938, 939*
<height>, *298*, *846*, *1002*
<join>, *506*, *613*, *965*
<width>, *298*, *846*, *1002*
@scribe
<handNote>, *354*, *747*
@script
<handNote>, *360*, *940*
<scriptStmt>, *228*, ***1121***
<seal>, *323*, ***1122***
<sealDesc>, *323*
<secFol>, *301*, ***1123***
<seg>, *187, 188*, *192, 193*, *231*, *239, 240*, *428*, *486–489*, *491*, *493*, *496*, *500*, *504*, *510, 511*, *513, 514*, *530*, *606–609*, *612, 613*, *615*, ***1124***
<segmentation>, *35*, ***1125***
@select
, *510, 511*
<seg>, *511*
<u>, *510*
<sense>, *254, 255*, *257–259*, *264–266*, *270*, *272*, *275, 276*, *282–284*, *287, 288*, *797*, *879*, *1085, 1086*, ***1126***, *1160*
@seq
<add>, *355*
<del>, *355*
<series>, *116*, *121*, *124*, *229*, *1089*, *1097*, ***1127***
<seriesStmt>, *21*, *28*, ***1128***
<set>, *200–202*, ***1129***
<setting>, *459*, *1069*, ***1130***
<settingDesc>, *459*, *1069*, ***1130***
<settlement>, *83*, *292–294*, *303–305*, *329*, *350*, *403*, *405*, *419–422*, *757*, *817*, *849, 850*, *959*, *995*, *1016, 1017*, *1020*, *1050*, *1057*, *1061, 1062*, *1102, 1103*, ***1131***
<sex>, *409*, ***1133***
@sex
<person>, *458*, *992*, *1050*, *1057*
<personGrp>, *1059*
<shift>, *233*, *237*, ***1133***
<sic>, *73, 74*, *248*, *321*, *345–349*, *360*, *362*, *806*, *824*, ***1135***
<signatures>, *301*, ***1136***
<signed>, *147–149*, *161*, *166*, *815*, *1066*, ***1137***
@since
<when>, *241–243*, *501*, *1187*, *1217*
@size
<graph>, *574–578*, *580*, *582*, *586*, *935*
<personGrp>, *1059*
<soCalled>, *67*, *143*, *236*, *381*, *528*, *764*, ***1138***
<socecStatus>, *409*, *458*, ***1139***

@social
<distinct>, *63*
@sort
<forename>, *397*, *399*
<surname>, *397*, *399*
@sortKey
<term>, *98, 99*, *1178*
<sound>, *223, 224*, *788*, ***1140***
<source>, *326*, *752*, *1088*, ***1141***
@source
<corr>, *349*
<normalization>, *35*, *1028*
<supplied>, *358*, *364*, *367*, *1162*
<writing>, *236*
<sourceDesc>, *21, 22*, *30, 31*, *50, 51*, *228*, *236*, *567*, *737*, *772*, *899*, ***1142***, *1176*
<sp>, *130–133*, *201–203*, *208*, *210, 211*, *213–221*, *223, 224*, *490*, *504*, *506*, *788*, *882*, *903*, *965*, *1071*, *1140*, ***1143***, *1146*, *1213*
<space>, *367*, ***1144***
@space
<distinct>, *63*
, *528, 529*, *535*, *1145*, ***1145***
<spanGrp>, *528, 529*, *535*, ***1145***
@spanTo
<addSpan>, *354*, *747*
<damageSpan>, *363, 364*, *831*
<delSpan>, *354, 355*, *841*
<index>, *98*
<speaker>, *130–133*, *203*, *208*, *210*, *213–221*, *223, 224*, *490*, *504*, *506*, *788*, *965*, *1140*, *1143*, ***1146***, *1213*
<specDesc>, *629*, ***1147, 1148***, *1150*
<specGrp>, *631*, ***1148***, *1149*
<specGrpRef>, *632*, ***1149***
<specList>, *629*, ***1150***
@split
<orth>, *287*
<pron>, *287*
<sponsor>, ***1151***
<stage>, *129–133*, *201–203*, *208*, *211–214*, *216–221*, *490*, *792*, *882*, *1014*, *1146*, ***1152***
<stamp>, *297*, ***1153***
@start
<schemaSpec>, *3–5*, *644*, *1120*
<surface>, *340*
<u>, *242–244*
<state>, *412*, *416*, *423*, ***1154***
@status
<availability>, *769*
<correction>, *51*
<del>, *356*
<stdVals>, *51*, ***1155***
<street>, *82, 83*, *751*, *1050*, ***1155, 1156***
<string>, *431*, *540, 541*, *544, 545*, *547*, *551*, *565*, *567*, ***1157***, *1204*
<stringVal>, ***1157***
<subc>, *257–259*, *263*, *934*, *948*, ***1158***, *1160*
<subst>, *353*, *355, 356*, ***1159***
@subtype
<anchor>, *608*, *610*
, *227*
<seal>, *323*, *1122*
<seg>, *487*, *489*
<summary>, ***1160***
<superEntry>, *253*, *255*, *259*, ***1160***
<supplied>, *347*, *358*, *364*, *367*, *830*, *1019*, ***1162***
<support>, *294*, *296*, *314*, *1060*, ***1163***, *1164*, *1216*
<supportDesc>, *294*, *314*, *1037*, *1060*, *1163*, ***1164***
<surface>, *333*, *335–337*, *339, 340*, *895*, ***1164***, *1226*
<surname>, *300*, *395–399*, *401*, *412*, *907*, *925*, *1024*, *1050*, *1056*, *1111*, ***1166***
<surrogates>, *328*, *748*, ***1167***
<syll>, *261*, ***1168***
<symbol>, *539, 540*, *544*, *546*, *548*, *552–555*, *563, 564*, *566, 567*, *775*, *820*, *892*, *894*, *914*, *919*, *954*, ***1168***, *1185*, *1204*, *1206–1208*
@synch
<anchor>, *241–244*, *499*
<seg>, *500*
<u>, *500*
<unclear>, *500*
<when>, *242*, *244, 245*

<table>, *436–439*, *813*, ***1169***
<tag>, *628*, *638*, ***1170***
@tag
<langKnown>, *409*, *458*, *974, 975*
@tags
<langKnowledge>, *409*, *458*, *974*
<tagsDecl>, *37, 38*, *1101*, *1172*, ***1172***
<tagUsage>, *9*, *37*, *39, 40*, *212*, *1024*, *1172*, ***1172***
@target
<catRef>, *44*, *48*, *795*, *1182*
<certainty>, *362*, *367*, *621–624*, *799–801*
<fsdLink>, *559*, *917*

<gloss>, *68*, *930*, *1179*
<locus>, *299*, *997*
<note>, *345*, *347, 348*, *389*, *471*, *620*
<oRef>, *278*
<pRef>, *279*
<ptr>, *44*, *88*, *90*, *126*, *161*, *166*, *269*, *274, 275*, *284*, *380*, *443*, *445*, *468, 469*, *475*, *478*, *480, 481*, *484*, *508*, *602*, *629*, *633*, *759*, *761, 762*, *813*, *993*, *1074*, *1222*
<ref>, *89*, *97*, *123*, *126*, *162*, *166*, *266*, *274, 275*, *282*, *289*, *326*, *445*, *471, 472*, *476–478*, *718*, *854*, *901*, *1019*, *1088*, *1091*
<respons>, *362*, *625*, *1106*
<specGrpRef>, *632*, *1149*
<witDetail>, *378*, *1220*
@targets
<alt>, *513, 514*, *755, 756*
<join>, *220*, *506–508*, *530*, *613*, *965, 966*
<link>, *196*, *243, 244*, *446*, *469*, *472–475*, *491*, *493*, *497, 498*, *500–502*, *505*, *512*, *531*, *535*, *556, 557*, *985*
@targFunc
<linkGrp>, *474*, *491*, *500–502*, *531*
<taxonomy>, *43*, *51*, *312*, *795*, *810*, ***1173***, *1182*
<tech>, ***1175***
<TEI>, *19*, *136*, *151*, *332*, *453*, *460*, *558, 559*, *567*, ***737***, *903*, *1052*, *1175*
<teiCorpus>, *19*, *453*, *460*, ***1175***
<teiHeader>, *18, 19*, *21, 22*, *50, 51*, *109*, *136*, *151*, *332*, *453*, *460*, *558, 559*, *567*, *737*, *903*, *1052*, *1175*, ***1176***
<term>, *8*, *61*, *68*, *81*, *90*, *93*, *98–100*, *139*, *318*, *320*, *857*, *930*, *938*, *958*, *970*, *1022*, *1030*, *1091*, ***1178, 1179***
@terminal
<metSym>, *193*, *197*, *1007*
<terrain>, *423*, ***1180***
<text>, *19*, *105*, *107*, *136*, *151, 152*, *155, 156*, *183–185*, *202*, *220*, *332*, *340*, *384*, *453*, *460*, *463*, *508*, *517*, *558, 559*, *737*, *903*, *937*, *1052*, *1175*, ***1181, 1182***
<textClass>, *51*, *810*, ***1182***
<textDesc>, *456, 457*, *794*, *1069*, ***1183***
<textLang>, *294*, *312, 313*, *1018, 1019*, ***1184***
<then>, *564*, *566, 567*, *820*, *914*, *954*, ***1185***
<time>, *86*, *430*, *432*, *459*, *707–710*, *1069*, ***1186***, *1235, 1236*
@time
<distinct>, *63*
<timeline>, *241–245*, *501*, ***1187***
<title>, *23, 24*, *28*, *30, 31*, *43*, *49–51*, *66*, *71*, *87*, *112–114*, *116–125*, *133*, *151*, *161, 162*, *205*, *228, 229*, *293, 294*, *298, 299*, *307*, *309–312*, *319*, *326*, *328*, *347*, *412*, *476*, *491*, *567*, *637*, *718*, *737*, *758*, *771, 772*, *774*, *782*, *802*, *821*, *899*, *901*, *989*, *997*, *1011*, *1015, 1016*, *1018, 1019*, *1042*, *1089*, *1097*, *1121*, *1127, 1128*, *1167*, *1176*, ***1189***, *1192*
<titlePage>, *39*, *105*, *152*, *164*, *200*, *862*, ***1190***
<titlePart>, *39*, *105*, *151, 152*, *164*, *200*, *218*, *862*, *866*, *979*, *1181*, *1190*, ***1191***
<titleStmt>, *21–24*, *30*, *49–51*, *113*, *567*, *737*, *772*, *802*, *899*, *1176*, ***1192***
<tns>, *286*, *1012*, *1053*, ***1193***
@to
<app>, *386, 387*
<arc>, *574*, *576–578*, *580*, *582*, *586*, *763*, *935*
<date>, *87*
<locus>, *298*, *309*, *312*, *317*, *319*, *979*, *1019*
<residence>, *393*
, *528, 529*, *535*, *1145*
<state>, *412*, *416*
<trait>, *1195*
<trailer>, *139*, *146*, *162*, *166*, *202*, *782*, *942*, *1071*, ***1194***
<trait>, *410*, ***1195***
@trans
<u>, *231*, *233*, *241*, *246*, *1199*
<tree>, *588, 589*, *591, 592*, ***1196***
<triangle>, *594*, ***1198***
@trunc
<numeric>, *541*, *1034*
@type
<ab>, *445*, *508*
<abbr>, *87*
<addName>, *397*, *399*, *747*
<altIdent>, *630*, *1010*
<altIdentifier>, *292–294*, *305*
<anchor>, *608*, *610*
<app>, *376–379*, *760*, *1220*
<biblScope>, *116–119*, *122*, *774*
<bloc>, *403*, *780*
<c>, *784*
<camera>, *787*
<castItem>, *206*, *791*
<cit>, *265–268*, *270*, *272, 273*, *276*, *278, 279*, *808*, *1036, 1037*, *1126*

<cl>, *524*, *809*
<classSpec>, *640*, *655*, *674, 675*, *812*
<colloc>, *263*, *818*
<constitution>, *456, 457*, *794*, *823*, *1069*, *1183*
<corr>, *348, 349*
<country>, *403*
<custEvent>, *327*, *828, 829*
<date>, *430–432*
<decoNote>, *320*, *837*
<del>, *77*, *240*, *247*, *840*
<derivation>, *456, 457*, *794*, *843*, *1069*, *1183*
<dimensions>, *294*, *298*, *846*, *1060*
<distinct>, *62*, *848*
<district>, *403*, *419*, *849, 850*, *995*
<div1>, *103, 104*, *107*, *131, 132*, *138, 139*, *141*, *143*, *152*, *162*, *202*, *208*, *213*, *219*, *220*, *446*, *489, 490*, *770*, *852*, *854*, *861*, *942*, *987*
<div2>, *103, 104*, *130–132*, *138, 139*, *141*, *143*, *208*, *219, 220*, *489, 490*, *852*, *854, 855*, *942*
<div3>, *103, 104*, *854, 855*, *857*
<div4>, *854*, *857*
, *7*, *66*, *96*, *100*, *138*, *140*, *143*, *145–149*, *152*, *157*, *161, 162*, *166, 167*, *186*, *190*, *193*, *200*, *209*, *227*, *246*, *384, 385*, *476, 477*, *492–494*, *496*, *498, 499*, *510–512*, *514*, *523*, *718*, *739*, *815*, *835*, *851*, *903*, *912, 913*, *985*, *1066*, *1213*
<divGen>, *100*, *160*, *861*
<domain>, *456, 457*, *794*, *868*, *1069*, *1183*
<eLeaf>, *601, 602*
<eTree>, *601, 602*
<entry>, *253*, *255*, *268*
<event>, *412*, *414*, *423*, *425*, *887*
<explicit>, *890*
<factuality>, *456, 457*, *794*, *896*, *1069*, *1183*
<filiation>, *901*
<floatingText>, *903*
<forename>, *395*, *397*
<forest>, *599*, *908*
<form>, *261, 262*, *266*, *276*, *279*, *282–284*, *286*, *288*, *789*, *981*, *1012*, *1048*, *1053*, *1193*
<fs>, *431*, *538*, *544, 545*, *547, 548*, *551*, *555*, *558, 559*, *564*, *567*, *914*
<fsDecl>, *558, 559*, *561*, *567*, *915–917*
<fsdLink>, *559*, *917*
<fw>, *368*, *920*
<geogName>, *403, 404*, *429*
<gram>, *263*, *932*
<graph>, *574–578*, *580*, *582*, *586*, *935*
<head>, *145*, *489*
<iType>, *262*, *953*
<ident>, *35*, *953*
<idno>, *27, 28*, *121*, *954*, *1016*, *1121*, *1128*, *1176*
<incipit>, *958*
<interaction>, *456, 457*, *794*, *961*, *1069*, *1183*
<interp>, *961*
<interpGrp>, *530*, *533, 534*, *962*
<join>, *220*
<kinesic>, *968*
<lbl>, *275*
<lg>, *128–130*, *152*, *185, 186*, *190, 191*, *193–195*, *201*, *216–220*, *379*, *882*, *984*
<link>, *505*, *512*
<linkGrp>, *196*, *243, 244*, *446*, *473, 474*, *491*, *493*, *497, 498*, *500–502*, *535*, *556, 557*, *985*
<list>, *7*, *65*, *84*, *91–94*, *96*, *148*, *166*, *212*, *349*, *477*, *625*, *942–944*, *963*, *970, 971*, *987, 988*, *1003*, *1081*, *1106*, *1129*, *1146*, *1199*
<listNym>, *990*
<listPerson>, *407*, *992*, *1133*
<listPlace>, *421, 422*, *992*
<location>, *420*
<m>, *526*, *998*, *1215*
<macroSpec>, *674*, *999*
<mapping>, *172*, *178–180*, *804, 805*, *999*
<measure>, *84, 85*, *315*, *1001*
<measureGrp>, *85*, *1002*
<metDecl>, *193*, *1006*
<move>, *214*, *1014*
<name>, *79–82*, *116*, *147*, *300*, *324, 325*, *392*, *395*, *400–402*, *404*, *412*, *414*, *416*, *458, 459*, *743*, *753*, *779*, *815*, *835, 836*, *1023*, *1041*, *1046*, *1127*, *1190*
<node>, *578*, *580*, *1027*
<note>, *95*, *276*, *469*, *471, 472*, *620*, *1030*
<num>, *83*, *1032*
<nym>, *428*
<oRef>, *270*, *278*, *1036*
<oVar>, *279*, *1037*
<orgName>, *400, 401*, *1043*
<orth>, *282–284*
<pause>, *1051*
<persName>, *395*, *399*
<phr>, *487*, *524*, *1060*
<place>, *418–422*, *426*, *995, 996*
<placeName>, *419, 420*, *995*
<population>, *425*, *1063*

<preparedness>, *456, 457*, *794*, *1068, 1069*, *1183*
<ptr>, *90*, *602*
<purpose>, *456, 457*, *794*, *1069*, *1078*, *1183*
<rdg>, *373*, *760*, *983*
<rdgGrp>, *376–378*, *760*, *1084*, *1220*
<re>, *276*
<recording>, *228, 229*, *1089, 1090*
<region>, *403*, *405*, *422*, *1096*
<relatedItem>, *123, 124*, *1097*
<relation>, *414*, *992*, *1050*, *1098, 1099*
<relationGrp>, *1099*
<restore>, *357*, *1107*
<roleName>, *398, 399*
<rs>, *46*, *78–80*, *203, 204*, *394*, *400*, *402*, *1054, 1055*, *1114*
<s>, *533*
<seal>, *323*, *1122*
<seg>, *187, 188*, *192, 193*, *240*, *428*, *486–489*, *510, 511*, *513*, *1124*
<settlement>, *403*, *405*, *419*, *995*, *1057*, *1131*
, *1145*
<spanGrp>, *529*, *535*, *1145*
<stage>, *130*, *132*, *203*, *212–214*, *216*, *219–221*, *490*, *792*, *882*, *1014*, *1146*, *1152*
<state>, *412*, *416*, *423*, *1154*
<surname>, *395–397*, *1056*, *1166*
<tag>, *1170*
<tech>, *1175*
<teiHeader>, *19*, *453*
<time>, *430*, *432*
<title>, *31*, *118*, *121*, *124*, *294*, *310–312*, *328*, *1019*, *1097*, *1121*, *1167*
<titlePage>, *200*
<titlePart>, *164*, *866*, *979*, *1190, 1191*
<trait>, *410*, *1195*
<usg>, *263*, *266*, *268*, *270*, *272, 273*, *275, 276*, *279*, *282–284*, *286*, *1085, 1086*, *1126*, *1203*
<valList>, *69*, *213*, *408*, *638*, *653*, *675*, *677*, *1211*
<w>, *487*, *526*, *998*, *1215*
<writing>, *236*
<xr>, *266*, *269*, *274, 275*, *282–284*, *1225*

<u>, *231*, *233–237*, *239–248*, *499, 500*, *510–513*, *535*, *623*, *1133*, *1145*, ***1199***
@ulx
<surface>, *335–337*, *339*, *895*, *1164*, *1226*
<zone>, *335–337*, *339*, *1226*
@uly
<surface>, *335–337*, *339*, *895*, *1164*, *1226*
<zone>, *335–337*, *339*, *1226*
<unclear>, *77*, *149*, *246*, *248*, *364, 365*, *499, 500*, ***1200***
<unicodeName>, *180*, *804*, *806*, ***1202***
@unit
<depth>, *842*
<dimensions>, *294*, *298*, *315*, *846*, *1060*
<gap>, *76*, *248*, *311*, *358*, *364*, *922*
<height>, *846*, *945*
<measure>, *84, 85*, *315*, *405*, *726*, *1001*
<measureGrp>, *85*, *1002*
<milestone>, *42*, *107, 108*, *1009*
<refState>, *42*, *111*, *1093*
<space>, *367*, *1144*
<timeline>, *241–243*, *501*, *1187*
<width>, *846*, *1218*
@uri
<equiv>, *69*, *885*
@url
<graphic>, *102*, *177*, *333*, *335–337*, *339, 340*, *443–445*, *476*, *895*, *898, 899*, *930*, *936*, *1164*, *1226*, *1243*
<moduleRef>, *644*, *1120*
@usage
<attDef>, *675*, *677*, *765, 766*, *875*
<elementSpec>, *633*, *875*
<language>, *47*, *51*, *975, 976*
<usg>, *263*, *266*, *268*, *270*, *272, 273*, *275, 276*, *279*, *282–284*, *286*, *1085, 1086*, *1126*, ***1203***

<val>, *49*, *51*, *484*, *628*, *765*, ***1209***
<valDesc>, *637*, ***1210***
<valItem>, *69*, *213*, *408*, *638*, *653*, *675*, *677*, *885*, ***1210***, *1211*
<valList>, *69*, *213*, *408*, *638*, *653*, *675*, *677*, *885*, ***1211***
<vAlt>, *550–554*, *563, 564*, *567*, *892, 893*, ***1204***, *1205*, *1208*
<value>, *172*, *177*, *179, 180*, *804*, *806*, *930*, ***1211***
@value
<age>, *754*
<binary>, *538*, *540*, *542*, *545*, *563, 564*, *566, 567*, *775, 776*, *820*, *892, 893*, *954*, *1185*, *1205*, *1208*
<case>, *789*
<eLeaf>, *868*
<gen>, *789*
<metSym>, *193*, *197*, *1006, 1007*
<mood>, *1012*, *1053*, *1193*
<node>, *582*, *586*

<num>, *83*, *1032*
<number>, *789*, *1012*, *1053*, *1193*
<numeric>, *431*, *540, 541*, *550*, *552*, *1034*
<per>, *1012*, *1053*, *1193*
<pos>, *789*
<sex>, *409*, *1133*
<symbol>, *539, 540*, *544*, *546*, *548*, *552–555*, *563, 564*, *566, 567*, *775*, *820*, *892*, *894*, *914*, *919*, *954*, *1168*, *1185*, *1204*, *1206–1208*
<tns>, *1012*, *1053*, *1193*
<variantEncoding>, *384–387*, ***1212***
@varSeq
<rdg>, *357*, *374*
<rdgGrp>, *377*
<vColl>, *547–551*, *553*, ***1204***, *1207*
<vDefault>, *563, 564*, *567*, *892, 893*, *954*, ***1205***, *1208*
@version
<application>, *44*, *761, 762*
<view>, *222–224*, *787, 788*, ***1213***
<vLabel>, *546*, ***1206***
<vMerge>, *553*, ***1207***
<vNot>, *552*, *554*, *565*, *567*, ***1207, 1208***
<vocal>, *231*, *233–235*, ***1214***
<vRange>, *563–565*, *567*, *892, 893*, *1205*, ***1208***

<w>, *340*, *487*, *526, 527*, *532–534*, *554*, *556, 557*, *610*, *613*, *615*, *998*, ***1215***
<watermark>, *296*, *314*, ***1216***
@weights
<alt>, *513, 514*, *755, 756*
<when>, *241–245*, *501*, *1187*, ***1217***
@when
<birth>, *393*, *458*, *779*, *1057*
<change>, *49*, *51*, *327*, *802*, *1088*, *1107*
<date>, *31*, *46*, *86, 87*, *121–123*, *149*, *203*, *205*, *228, 229*, *294*, *328*, *412*, *430–433*, *709, 710*, *758*, *782*, *827*, *832*, *834*, *1011*, *1066*, *1076*, *1089*, *1176*
<death>, *836*
<docDate>, *39*
<event>, *412*, *423*, *425*, *887*
<population>, *425*, *1063*
<time>, *86*, *430*, *432*, *709, 710*, *1186*
@when-iso
<date>, *707*, *709*
<time>, *707, 708*
@where
<move>, *214*, *1014*
@who
<change>, *49*, *802*
<kinesic>, *241*
<move>, *214*, *1014*
<q>, *508*
<said>, *65*
<setting>, *459*
<sp>, *132*, *210, 211*, *214*, *1146*
<u>, *231*, *233–236*, *240, 241*, *243–246*, *248*, *499, 500*, *623*, *1133*, *1199*
<vocal>, *231*, *233, 234*
<writing>, *236*
<width>, *294*, *298*, *315*, *846*, *1002*, *1060*, ***1218***
@width
<graphic>, *1243*
<wit>, *376, 377*, *379*, *760*, ***1219***
@wit
<lem>, *373*, *376–378*, *383*, *385–387*, *760*, *972*, *983*, *1084*, *1220, 1221*
<rdg>, *350*, *372–374*, *376–379*, *382–388*, *760*, *971, 972*, *983*, *1083, 1084*, *1219–1221*
<witDetail>, *378*, *1220*
<witDetail>, *378*, ***1220***
<witEnd>, ***1221***
@withId
<tagUsage>, *40*, *1024*, *1172*
<witness>, *380–382*, *993*, ***1222***
<witStart>, *383*, ***1221***
<writing>, *236*, ***1223***
@writtenLines
<layout>, *317*, *978*

<xr>, *266*, *269*, *274, 275*, *282–284*, *886*, *973*, ***1225***

<zone>, *335–337*, *339*, ***1226***