5 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 it has four major parts:

- 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 a file description are described in section 5.2 *The File Description* below.
- 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 5.3 *The Encoding Description* below.
- 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 5.4 *The Profile Description* below.
- 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 5.5 *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) will 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 *additional tagsets*, which may be invoked as extensions as needed. For more details of this extension mechanism, see chapter 3.2 *Core, Base, and Additional Tag Sets*; the header extensions are fully described in chapter 23 *Language Corpora*, which should be read in conjunction with the present 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 5.6 *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. Recommendations relevant to the use of TEI headers as free-standing documents, for interchange among libraries, data archives, and similar institutions may be found in chapter 24 *The Independent Header*.

5.1 Organization of the TEI Header

5.1.1 The TEI Header and Its Components

The <teiHeader> element should be clearly distinguished both from the *prolog*, which comprises either the XML declaration or the SGML declaration, and the document type declaration (see chapter 2 *A Gentle Introduction to XML*); and from the *front matter* of the text itself (for which see section 7.4 *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> supplies the descriptive and declarative information making up an "electronic title page" prefixed to every TEI-conformant text. Attributes include:

type specifies the kind of document to which the header is attached.

Sample values include:

`text` the header is attached to a single text.

`corpus` the header is attached to a corpus.

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

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

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

<profileDesc> 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> summarizes the revision history for a file.

Of these, only the <fileDesc> element is required in all TEI headers; the others are optional. The full form of a TEI header is 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 individual components in the corpus or collection, or once for all at the level of the corpus or collection itself, or at both levels. More details concerning the tagging of composite texts are given in section 23 *Language Corpora*, which should be read in conjunction with the current chapter. An optional type attribute may also be supplied on the <teiHeader> element to indicate whether the header applies to a corpus or a single text. A corpus may thus take the form:

```
<teiCorpus.2>
   <teiHeader type='corpus'>
      <!-- header for corpus-level information -->
   </teiHeader>
   <TEI.2>
      <teiHeader type='text'>
         <!-- header for text-level information -->
      </teiHeader>
      <text> <!-- ... --> </text>
   </TEI.2>
   <TEI.2>
      <teiHeader type='text'> <!-- ... --> </teiHeader>
      <text> <!-- ... --> </text>
```

```
   </TEI.2>
   <!-- etc. -->
</teiCorpus.2>
```

The tags required for the TEI header are defined in the DTD file teihdr2.dtd which first defines the <teiHeader> element:

```
<!-- 5.1.1: The TEI Header-->
<!--teihdr2.dtd  Tags for TEI Header.-->
<!--Text Encoding Initiative Consortium:
Guidelines for Electronic Text Encoding and Interchange.
Document TEI P4, 2002.
Copyright (c) 2002 TEI Consortium. Permission to copy in any form
is granted, provided this notice is included in all copies.
These materials may not be altered; modifications to these DTDs should
be performed only as specified by the Guidelines, for example in the
chapter entitled 'Modifying the TEI DTD'
These materials are subject to revision by the TEI Consortium. Current versions
are available from the Consortium website at http://www.tei-c.org-->
<!ELEMENT teiHeader %om.RR; (fileDesc, encodingDesc*, profileDesc*,
revisionDesc?)>
<!ATTLIST teiHeader
      %a.global;
      type CDATA "text"
      creator CDATA #IMPLIED
      status (new | update) "new"
      date.created %ISO-date; #IMPLIED
      date.updated %ISO-date; #IMPLIED
      TEIform CDATA 'teiHeader'  >
<!--continued in 5.1.1: -->
<!-- end of 5.1.1-->
```

Then it defines the rest of the header elements, embedding the DTD fragments found later in this chapter:

```
<!-- 5.1.1: -->
<!--declarations from 5.2: The file description inserted here -->
<!--declarations from 5.2.7: The source description inserted here -->
<!--declarations from 5.3: The encoding description inserted here -->
<!--declarations from 5.4: The profile description inserted here -->
<!--declarations from 5.5: The Revision Description inserted here -->
<!-- end of 5.1.1-->
```

5.1.2 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 with the <p> tag. 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 6.1 *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. [64] 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.

[64] 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 5.7 *Note for Library Cataloguers*.

declarations Elements whose names end with the suffix 'Decl' (e.g. <subjectDecl>, <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. 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 23.3 *Associating Contextual Information with a Text*.

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

5.2 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 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 6.10 *Bibliographic Citations and References*). See further section 5.7 *Note for Library Cataloguers*.

The bibliographic description of the electronic text (not its source) is given in the mandatory <fileDesc> element:

<fileDesc> 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 5.2.1 *The Title Statement* to 5.2.6 *The Notes Statement* below. These elements are listed below in the order in which they must be given within the <fileDesc> element.

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

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

<extent> describes the approximate size of the electronic text as stored on some carrier medium, specified in any convenient units.

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

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

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

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

A file description containing all possible subelements has the following structure:

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

```
    <!-- remainder of TEI Header here -->
</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>
    <!-- remainder of TEI Header here -->
</teiHeader>
```

The <fileDesc> itself has the following formal definition:

```
<!-- 5.2: The file description-->
<!ELEMENT fileDesc %om.RR; (titleStmt, editionStmt?, extent?,
          publicationStmt, seriesStmt?, notesStmt?,
          sourceDesc+ ) >
<!ATTLIST fileDesc
     %a.global;
     TEIform CDATA 'fileDesc'  >
<!--declarations from 5.2.1: The title statement inserted here -->
<!--declarations from 5.2.2: The edition statement inserted here -->
<!--declarations from 5.2.3: The extent statement inserted here -->
<!--declarations from 5.2.4: The publication statement inserted here -->
<!--declarations from 5.2.5: The series statement inserted here -->
<!--declarations from 5.2.6: The notes statement inserted here -->
<!-- end of 5.2-->
```

5.2.1 The Title Statement

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

<titleStmt> 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 title of a work, whether article, book, journal, or series, including any alternative titles or subtitles. Attributes include:

level (bibliographic level (or class) of title) indicates whether this is the title of an article, book, journal, series, or unpublished material.

Legal values are:

- `a` analytic title (article, poem, or other item published as part of a larger item)
- `m` monographic title (book, collection, or other item published as a distinct item, including single volumes of multi-volume works)
- `j` journal title
- `s` series title
- `u` title of unpublished material (including theses and dissertations unless published by a commercial press)

type (type of title) classifies the title according to some convenient typology.

Sample values include:

- `main` main title
- `subordinate` subtitle, title of part
- `parallel` alternate title, often in another language, by which the work is also known
- `abbreviated` abbreviated form of title

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

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

<respStmt> supplies a statement of responsibility for someone responsible 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> contains a phrase describing the nature of a person's intellectual responsibility.

<name> contains a proper noun or noun phrase. Attributes include:

type indicates the type of the object which is being named by the phrase.

Values Values such as person, place, institution, product, acronym.

The `<title>` element contains the chief name of the file, including any alternative title or subtitles it may have. It may be repeated, if the file 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 also be derived from the latter, but that it should be clearly distinguishable from it. For example, do not call the computer file "A Sanskrit-English Dictionary, based upon the St. Petersburg Lexicons". Call it, rather, "Sanskrit-English Dictionary, based upon the St. Petersburg Lexicons: a machine readable transcription". If you wish to retain some or all of the title of the source text in the title of the computer file, then introduce one of the following phrases:

- [title of source]: a machine readable transcription.
- [title of source]: electronic edition.
- A machine readable version of: [title of source].

This will distinguish the computer file from the source text in citations and in catalogues which contain descriptions of both types of material.

The computer file will almost certainly 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. For these reasons, these Guidelines strongly recommend that such names should *not* be used as the `<title>` for any computer file.

Helpful guidance on the formulation of useful descriptive titles in difficult cases may be found in the Anglo-American Cataloguing Rules[65] (AACR 2), chapter 25, or in equivalent national-level bibliographical documentation.

The specialized elements `<author>`, `<sponsor>`, `<funder>`, and `<principal>`, and the more general `<respStmt>` provide the *statements of responsibility* which identify the persons responsible for the intellectual or artistic content of an item and any corporate bodies from which it emanates.

Any number of statements of responsibility may occur within the title statement. At a minimum, identify the author of the text and the creator of the machine-readable file. If the bibliographic description is for a corpus, identify the creator of the corpus. These identifications are mandatory when applicable, though not enforceable by the parser. 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

[65] Michael Gorman and Paul W. Winkler, eds., *Anglo-American Cataloguing Rules*, Second Edition (Chicago: American Library Association; London: Library Association; Ottawa: Canadian Library Association, 1978).

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.[66]

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>

<titleStmt>
  <title>Yogadar&sacute;anam (arth&amacr;t
         yogas&umacr;trap&macr;&tdot;ha&hdot;):
         a machine readable transcription.</title>
  <title>The Yogas&umacr;utras of Pata&ntilde;jali:
         a machine readable transcription.</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>
```

The formal definition of the <titleStmt> element and its constituents is as follows:

```
<!-- 5.2.1: The title statement-->
<!ELEMENT titleStmt %om.RO; ((title+, (author | editor
| sponsor | funder | principal
| respStmt)*))>
<!ATTLIST titleStmt
      %a.global;
      TEIform CDATA 'titleStmt'  >
<!ELEMENT sponsor %om.RO; %phrase.seq; >
<!ATTLIST sponsor
      %a.global;
      TEIform CDATA 'sponsor'  >
<!ELEMENT funder %om.RO;  %phrase.seq; >
<!ATTLIST funder
      %a.global;
      TEIform CDATA 'funder'  >
<!ELEMENT principal %om.RO; %phrase.seq;>
<!ATTLIST principal
      %a.global;
      TEIform CDATA 'principal'  >
<!--The TITLE, AUTHOR, NAME, RESPSTMT, and RESP elements are
declared in file teicore2.dtd, not here.-->
<!-- end of 5.2.1-->
```

[66] 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.

5.2.2 The Edition Statement

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

<editionStmt> 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> describes the particularities of one edition of a text.

<respStmt> supplies a statement of responsibility for someone responsible 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> contains a proper noun or noun phrase. Attributes include:

type indicates the type of the object which is being named by the phrase.

Values Values such as person, place, institution, product, acronym.

<resp> 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. The addition of new information (e.g. a linguistic analysis expressed in part-of-speech tagging, sound or graphics, referential links to external datasets) almost always does constitute a new edition.

Clearly, there are always border line 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 machine-readable 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 5.5 *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 6.10 *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>
```

The formal definition of the <editionStmt> element is as follows:

```
<!-- 5.2.2: The edition statement-->
<!ELEMENT editionStmt %om.RO; ( (edition, respStmt*) | p+ )>
<!ATTLIST editionStmt
      %a.global;
      TEIform CDATA 'editionStmt'  >
<!ELEMENT edition %om.RO; %phrase.seq;>
<!ATTLIST edition
      %a.global;
      TEIform CDATA 'edition'  >
<!-- end of 5.2.2-->
```

5.2.3 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 the electronic text as stored on some carrier medium, 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)

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 3.5" High Density Diskettes</extent>
```

The <extent> element has the following formal declaration:

```
<!-- 5.2.3: The extent statement-->
<!ELEMENT extent %om.RO; %phrase.seq;>
<!ATTLIST extent
      %a.global;
      TEIform CDATA 'extent'  >
<!-- end of 5.2.3-->
```

5.2.4 Publication, Distribution, etc.

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

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

It may contain either a simple prose description, or groups of the elements described 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> 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:

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

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

<idno> supplies any standard or non-standard number used to identify a bibliographic item. Attributes include:

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

Values A name or abbreviation indicating what type of identifying number is given (e.g. ISBN, LCCN).

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

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

Legal values are:

`free` the text is freely available.

`unknown` the status of the text is unknown.

`restricted` the text is not freely available.

<date> contains a date in any format. Attributes include:

calendar indicates the system or calendar to which the date belongs.

Values Recommended values include: *Gregorian*, *Julian*, *Roman*, *Mosaic*, *Revolutionary*, *Islamic*.

value gives the value of the date in some standard form, usually yyyy-mm-dd.

Values Any string representing a date in standard format; recommended form is ISO 8601:2000 5.2.1.1 Complete representation, extended format (yyyy-mm-dd)

certainty indicates the degree of precision to be attributed to the date.

Values Any appropriate value, e.g. *ca.*, *approx*, *after*, *before*.

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 tagsets may be used for the encoding of names, dates, and addresses, as further described in section 6.4 *Names, Numbers, Dates, Abbreviations, and Addresses* and chapter 20 *Names and Dates*.

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>
```

The publication statement and its components are formally defined as follows:

```
<!-- 5.2.4: The publication statement-->
<!ELEMENT publicationStmt %om.RO;
( ( p, (%m.Incl;)*)+
  | ( (publisher | distributor | authority | pubPlace | address | idno
  | availability | date ), (%m.Incl;)*)+ )>
<!ATTLIST publicationStmt
      %a.global;
      TEIform CDATA 'publicationStmt'  >
<!ELEMENT distributor %om.RO; %phrase.seq;>
<!ATTLIST distributor
      %a.global;
      TEIform CDATA 'distributor'  >
<!ELEMENT authority %om.RO; %phrase.seq;>
<!ATTLIST authority
      %a.global;
      TEIform CDATA 'authority'  >
<!ELEMENT idno %om.RO; (#PCDATA)>
<!ATTLIST idno
      %a.global;
      type CDATA #IMPLIED
      TEIform CDATA 'idno'  >
<!ELEMENT availability %om.RO; (p+)>
<!ATTLIST availability
      %a.global;
      status ( free | unknown | restricted ) "unknown"
      TEIform CDATA 'availability'  >
<!--The PUBLISHER, PUBPLACE, and ADDRESS elements
are defined in file teicore2.dtd.-->
<!-- end of 5.2.4-->
```

5.2.5 The Series Statement

The <seriesStmt> element is the fifth component of the <fileDesc> element and is optional.
<seriesStmt> 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 title of a work, whether article, book, journal, or series, including any alternative titles or subtitles. Attributes include:

level (bibliographic level (or class) of title) indicates whether this is the title of an article, book, journal, series, or unpublished material.

Legal values are:

- `a` analytic title (article, poem, or other item published as part of a larger item)
- `m` monographic title (book, collection, or other item published as a distinct item, including single volumes of multi-volume works)
- `j` journal title
- `s` series title
- `u` title of unpublished material (including theses and dissertations unless published by a commercial press)

type (type of title) classifies the title according to some convenient typology.

Sample values include:

- `main` main title
- `subordinate` subtitle, title of part
- `parallel` alternate title, often in another language, by which the work is also known
- `abbreviated` abbreviated form of title

<idno> supplies any standard or non-standard number used to identify a bibliographic item. Attributes include:

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

Values A name or abbreviation indicating what type of identifying number is given (e.g. ISBN, LCCN).

<respStmt> supplies a statement of responsibility for someone responsible 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> contains a phrase describing the nature of a person's intellectual responsibility.

<name> contains a proper noun or noun phrase. Attributes include:

type indicates the type of the object which is being named by the phrase.

Values Values such as person, place, institution, product, acronym.

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>
```

The series statement has the following formal definition:

```
<!-- 5.2.5: The series statement-->
<!ELEMENT seriesStmt %om.RO; ( (title+, (idno |  respStmt)*)
| p+ )>
<!ATTLIST seriesStmt
      %a.global;
      TEIform CDATA 'seriesStmt'  >
<!-- end of 5.2.5-->
```

Its components are all defined elsewhere.

5.2.6 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> 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. Attributes include:

type describes the type of note.

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) indicates who is responsible for the annotation: author, editor, translator, etc.

Sample values include:

- auth[or] note originated with the author of the text.
- ed[itor] note added by the editor of the text.
- comp[iler] note added by the compiler of a collection.
- tr[anslator] note added by the translator of a text.
- transcr[iber] note added by the transcriber of a text into electronic form.
- (initials) note added by the individual indicated by the initials.

place indicates where the note appears in the source text.

Sample values include:

- foot note appears at foot of page.
- end note appears at end of chapter or volume.
- inline note appears as a marked paragraph in the body of the text.
- left note appears in left margin.
- right note appears in right margin.
- interlinear note appears between lines of the text.
- app[aratus] note appears in the apparatus at the foot of the page.

anchored indicates whether the copy text shows the exact place of reference for the note.

Legal values are:

- yes copy text indicates the place of attachment for the note.
- no copy text indicates no place of attachment for the note.

target indicates the point of attachment of a note, or the beginning of the span to which the note is attached.

Values reference to the ids of element(s) which begin at the location in question (e.g. the id of an <anchor> element).

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.

Values reference to the id(s) of element(s) which *end* at the location(s) in question, or to an empty element at the point in question.

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 5.4 *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 5.4 *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 5.2.7 *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 5.2.4 *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: 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)
- lists of related publications, either describing the source itself, or concerned with the creation or use of the machine-readable file, e.g. "Texts used in *Computation into Criticism* (Oxford, 1987)"

Each such item of information should be tagged using the general-purpose <note> element, which is described in section 6.8 *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>
```

The notes statement has the following formal definition:

```
<!-- 5.2.6: The notes statement-->
<!ELEMENT notesStmt %om.RO; (note+)>
<!ATTLIST notesStmt
```

```
      %a.global;
      TEIform CDATA 'notesStmt'  >
<!--The NOTE element is defined with the core tags.-->
<!-- end of 5.2.6-->
```

5.2.7 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> supplies a bibliographic description of the copy text(s) from which an electronic text was derived or generated.

The <sourceDesc> element may contain a simple prose description, or, more usefully, a bibliographic citation of some kind specifying the provenance of the text. For written or printed sources, the source should be described in the same way as any other bibliographic citation, using one of the following elements:

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

<biblStruct> contains a structured bibliographic citation, in which only bibliographic subelements appear and in a specified order.

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

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

These elements are described in more detail in section 6.10 *Bibliographic Citations and References* .

When the header describes a transcription of spoken material, 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> contains a citation giving details of the script used for a spoken text.

<recordingStmt> describes a set of recordings used in transcription of a spoken text.

Full descriptions of these elements and their contents are given in section 5.2.9 *Computer Files Composed of Transcribed Speech.*

The <sourceDesc> element may contain a mixture of one or more of the above elements, as in the following examples:

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

<sourceDesc>
   <p>No source: created in machine-readable form.</p>
</sourceDesc>

<sourceDesc>
  <biblStruct lang='FR'>
    <monogr>
      <author>Eug&egrave;ne Sue</author>
      <title>Martin, l'enfant trouv&eacute;</title>
      <title type='sub'>M&eacute;moires d'un valet de chambre</title>
      <imprint>
        <pubPlace>Bruxelles et Leipzig</pubPlace>
        <publisher>C. Muquardt</publisher>
        <date value="1846">1846</date>
      </imprint>
    </monogr></biblStruct>
</sourceDesc>
```

The source description itself has the following formal definition:

```
<!-- 5.2.7: The source description-->
<!ELEMENT sourceDesc %om.RR; (p | bibl | biblFull | biblStruct
                            | listBibl | scriptStmt | recordingStmt )+ >
<!ATTLIST sourceDesc
      %a.global;
      %a.declarable;
      TEIform CDATA 'sourceDesc'  >
<!--declarations from 5.2.9: Script statement and recording statement inserted here -->
<!-- end of 5.2.7-->
```

5.2.8 Computer Files Derived from Other Computer Files

If a machine-readable text (call it B) is based not on a printed source but upon another machine-readable text (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 (see section 6.10 *Bibliographic Citations and References*).

profileDesc A's <profileDesc> should be copied into B's, in principle unchanged.

encodingDesc A's coding practice may or (more likely) may not be the same as B's. Since the object of the coding 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 electronic 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.

5.2.9 Computer Files Composed of Transcribed Speech

Where an electronic text 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 element:

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

<recordingStmt> describes a set of recordings used in transcription of a spoken text.

<recording> details of an audio or video recording event used as the source of a spoken text, either directly or from a public broadcast. Attributes include:

type the kind of recording.

Legal values are:

`audio` audio recording

`video` audio and video recording

dur the original duration of the recording.

Values Include the units, e.g. 30 min.

<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.

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 23 *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 23.3 *Associating Contextual Information with a Text*.

```
<sourceDesc>
  <scriptStmt id='CNN12'>
    <bibl>
      <author>CNN Network News</author>
      <title>News headlines</title>
      <date value="1991-06-12">12 Jun 91</date>
    </bibl>
  </scriptStmt>
  <!-- this script statement might be used to document the parts
       of a spoken transcript which included a news broadcast -->
  <!-- possibly other script statements or recording statements follow -->
</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 23.3 *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 a prose description, within which such items as statements of responsibility, names, places and dates should be identified using the appropriate phrase level tags. The <recording> element takes two additional attributes, as indicated above: type is used to specify the kind of recording concerned and dur to specify its length.

In addition, descriptive information relating to the kind of recording equipment used should be specified using the <equipment> element. Where a recording is taken from a public broadcast, details of the broadcast should be given using the <broadcast> element described further below. Specialized collections may wish to add further sub-elements to these major components. Note however that this element 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 23.2.3 *The Setting Description* and 23.2.2 *The Participants Description* below.

```
<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>

<recording type='audio' dur="30 min">
  <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>
```

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, directors, producers, etc.) should be specified using the <respStmt> or <editor> element with an appropriate <resp> (see further section 6.10 *Bibliographic Citations and References*).

```
<recording type='audio' dur="10 min">
  <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 value="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='100'>
   <broadcast>
         <!-- details of broadcast -->
      <recording>
         <!-- details of broadcast recording -->
      </recording>
   </broadcast>
</recording>
```

Formal definitions for the elements discussed in this section are as follows:

```
<!-- 5.2.9: Script statement and recording statement-->
<!ELEMENT scriptStmt %om.RR; (p+ | bibl | biblFull | biblStruct)>
<!ATTLIST scriptStmt
      %a.global;
      %a.declarable;
      TEIform CDATA 'scriptStmt'  >
<!ELEMENT recordingStmt %om.RR; (p+ | recording+ )>
<!ATTLIST recordingStmt
      %a.global;
      TEIform CDATA 'recordingStmt'  >
<!ELEMENT recording %om.RR; (p+ | (respStmt | equipment | broadcast |
date)*)>
<!ATTLIST recording
      %a.global;
      %a.declarable;
      type (audio | video) "audio"
      dur CDATA #IMPLIED
      TEIform CDATA 'recording'  >
<!ELEMENT equipment %om.RO; (p+)>
<!ATTLIST equipment
      %a.global;
      %a.declarable;
      TEIform CDATA 'equipment'  >
<!ELEMENT broadcast %om.RR; (p+ | bibl | biblStruct | biblFull | recording)>
<!ATTLIST broadcast
      %a.global;
      %a.declarable;
      TEIform CDATA 'broadcast'  >
<!-- end of 5.2.9-->
```

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

5.3 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> documents the relationship between an electronic text and the source or sources from which it was derived.

The content of the encoding description may be a prose description, or it may contain elements from the following list, in the order given:

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

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

<tagsDecl> provides detailed information about the tagging applied to an SGML or XML document.

<refsDecl> specifies how canonical references are constructed for this text. Attributes include:

doctype identifies the *document type* within which this reference declaration is used.
Values must be the name of a document type

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

<fsdDecl> identifies the feature system declaration which contains definitions for a particular type of feature structure. Attributes include:

type identifies the type of feature structure documented in the FSD; this will be the value of the type attribute on at least one feature structure.
Values any string of characters.

fsd (feature-system declaration) specifies the external entity containing the feature system declaration; an entity declaration in the document's DTD subset must associate the entity name with a file on the system.
Values a valid external entity name

<metDecl> 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>). Attributes include:

type indicates whether the notation conveys the abstract metrical form, its actual prosodic realization, or the rhyme scheme, or some combination thereof.
Values One or more of the three attribute names met, real, or rhyme, separated by whitespace.

pattern specifies a regular expression defining any value that is legal for this notation.
Values the value must be a valid expression for the PATTERN keyword as defined in the TEI extended pointer notation (see section 14.2.2.14 *The PATTERN Keyword*).

<variantEncoding> declares the method used to encode text-critical variants. Attributes include:

method indicates which method is used to encode the apparatus of variants.
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.

location indicates whether the apparatus appears within the running text or external to it.
Legal values are:

`internal` apparatus appears within the running text.

`external` apparatus appears outside the base text.

Each of these elements is further described and formally defined in the appropriate section below. The encoding description itself is defined as follows:

```
<!-- 5.3: The encoding description-->
<!ELEMENT encodingDesc %om.RR; (projectDesc*, samplingDecl*,
          editorialDecl*, tagsDecl?, refsDecl*,
          classDecl*, metDecl*, fsdDecl*,
```

```
            variantEncoding*, p* )>
<!ATTLIST encodingDesc
      %a.global;
      TEIform CDATA 'encodingDesc'  >
<!--declarations from 5.3.1: The project description inserted here -->
<!--declarations from 5.3.2: The sampling declaration inserted here -->
<!--declarations from 5.3.3: The editorial practices declaration inserted here -->
<!--declarations from 5.3.4: Tag usage and rendition declarations inserted here -->
<!--declarations from 5.3.5.3: The reference scheme declaration inserted here -->
<!--declarations from 5.3.6: The classification declaration inserted here -->
<!--declarations from 5.3.7: The FSD declaration inserted here -->
<!--declarations from 5.3.8: Metrical Notation Declaration inserted here -->
<!--declarations from 5.3.9: Variant-Encoding Declaration inserted here -->
<!-- end of 5.3-->
```

5.3.1 The Project Description

The <projectDesc> element is the first of the nine optional subdivisions of the <encodingDesc> element. It may be used to describe, in prose, the purpose for which the electronic file was encoded, 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> 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>
```

This element has the following formal declaration:

```
<!-- 5.3.1: The project description-->
<!ELEMENT projectDesc %om.RO; (p+)>
<!ATTLIST projectDesc
      %a.global;
      %a.declarable;
      TEIform CDATA 'projectDesc'  >
<!-- end of 5.3.1-->
```

5.3.2 The Sampling Declaration

The <samplingDecl> element is the second of the nine optional subdivisions of the <encodingDesc> element. It contains a prose description of the rationale and methods used in sampling texts, for example to create a representative corpus.

<samplingDecl> 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 23.3 *Associating Contextual Information with a Text*. This element has the following formal declaration:

```
<!-- 5.3.2: The sampling declaration-->
<!ELEMENT samplingDecl %om.RO; (p+)>
<!ATTLIST samplingDecl
      %a.global;
      %a.declarable;
      TEIform CDATA 'samplingDecl'  >
<!-- end of 5.3.2-->
```

5.3.3 The Editorial Practices Declaration

The <editorialDecl> element is the third of the nine optional subdivisions of the <encodingDesc> element. It is used to provide details of the editorial practices applied during the encoding of a text.

<editorialDecl> 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 the following specialized elements:

<correction> states how and under what circumstances corrections have been made in the text. Attributes include:

status indicates the degree of correction applied to the text.

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.

method indicates the method adopted to indicate corrections within the text.

Legal values are:

`silent` corrections have been made silently

`tags` corrections have been represented using editorial tags

<normalization> indicates the extent of normalization or regularization of the original source carried out in converting it to electronic form. Attributes include:

source indicates the authority for any normalization carried out.

Values Should really be a bibliographic reference of some kind

method indicates the method adopted to indicate normalizations within the text.

Legal values are:

`silent` normalization made silently

`tags` normalization represented using editorial tags

<quotation> specifies editorial practice adopted with respect to quotation marks in the original. Attributes include:

marks indicates whether or not quotation marks have been retained as content within the text.

Legal values are:

`none` no quotation marks have been retained

`some` some quotation marks have been retained

`all` all quotation marks have been retained

form specifies how quotation marks are indicated within the text.

Legal values are:

`data` quotation marks are retained as data.

`rend` the rendition attribute is consistently used to indicate the form of quotation marks.

`std` use of quotation marks has been standardized.

`nonstd` quotation marks are represented inconsistently.

`unknown` use of quotation marks is unknown.

<hyphenation> summarizes the way in which hyphenation in a source text has been treated in an encoded version of it. Attributes include:

eol indicates whether or not end-of-line hyphenation has been retained in a text.

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.

`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.

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

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

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

Some of these 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 listed below, in the same order as the list above.

<correction> 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 6.5 *Simple Editorial Changes*? What principles have been adopted with respect to omissions, truncations, dubious corrections, alternate readings, false starts, repetitions, etc.?

<normalization> 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 6.5 *Simple Editorial Changes*? What authority was used for the regularization? Also, what principles were used when normalizing dates or numbers to provide the standard values for the value attribute described in sections 6.4.3 *Numbers and Measures* and 6.4.4 *Dates and Times* and what format used for them?

<quotation> 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? Was the rendition attribute used to record the specific appearance of any quotation marks removed from the text? 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> 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> 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> What standardization methods underly any standardized values supplied for numeric values or dates? If the value attribute described in section 6.4.4 *Dates and Times* has been used, in what format are its values presented?

<interpretation> Has any analytic or 'interpretive' information been provided — that is, information which is felt to be non-obvious, contentious, or subject to disagreement? If so, how was it generated? How was it encoded? If feature-structure analysis has been used, are <fsdDecl> elements (section 5.3.7 *The 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. A simple example follows:

```
<editorialDecl id="e2">
  <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="W9">
    <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 represented by entity reference ODQ; all closing
      quotation marks represented by entity reference CDQ.</p>
  </quotation>
</editorialDecl>
```

These elements are formally defined as follows:

```
<!-- 5.3.3: The editorial practices declaration-->
<!ELEMENT editorialDecl %om.RO; ( p+ | ((correction | normalization
       | quotation | hyphenation | interpretation
       | segmentation | stdVals)+, p*))>
<!ATTLIST editorialDecl
      %a.global;
      %a.declarable;
      TEIform CDATA 'editorialDecl'  >
<!ELEMENT correction %om.RO; (p+)>
<!ATTLIST correction
      %a.global;
      %a.declarable;
      status (high | medium | low | unknown) "unknown"
      method (silent | tags) "silent"
      TEIform CDATA 'correction'  >
<!ELEMENT normalization %om.RO; (p+)>
<!ATTLIST normalization
      %a.global;
      %a.declarable;
      source CDATA #IMPLIED
      method ( silent | tags ) "silent"
      TEIform CDATA 'normalization'  >
<!ELEMENT quotation %om.RO; (p+)>
<!ATTLIST quotation
      %a.global;
      %a.declarable;
      marks ( none | some | all ) "all"
      form (data | rend | std | nonstd | unknown) "unknown"
      TEIform CDATA 'quotation'  >
<!ELEMENT hyphenation %om.RO; (p+)>
<!ATTLIST hyphenation
```

```
      %a.global;
      %a.declarable;
      eol ( all | some | none ) "some"
      TEIform CDATA 'hyphenation'  >
<!ELEMENT segmentation %om.RO; (p+)>
<!ATTLIST segmentation
      %a.global;
      %a.declarable;
      TEIform CDATA 'segmentation'  >
<!ELEMENT stdVals %om.RO; (p+)>
<!ATTLIST stdVals
      %a.global;
      %a.declarable;
      TEIform CDATA 'stdVals'  >
<!ELEMENT interpretation %om.RO; (p+)>
<!ATTLIST interpretation
      %a.global;
      %a.declarable;
      TEIform CDATA 'interpretation'  >
<!-- end of 5.3.3-->
```

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 23.3 *Associating Contextual Information with a Text*.

5.3.4 The Tagging Declaration

The <tagsDecl> element is the fourth of the nine optional subdivisions of the <encodingDesc> element. It is used to record the following information about the tagging used within a particular text:

- 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 definition for the default rendition applying to all instances of an element, unless otherwise stated by the global rend attribute.

This information is conveyed by the following elements:

<rendition> supplies information about the intended rendition of one or more elements.

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

occurs specifies the number of occurrences of this element within the text.
Values an integer number greater than zero

ident specifies the number of occurrences of this element within the text which bear a distinct value for the global id attribute.
Values an integer number greater than zero

render specifies the identifier of a <rendition> element which defines how this element is to be rendered.
Values an identifier specified as the value of the id attribute on some <rendition> element in the current document.

The <tagsDecl> element consists of an optional sequence of <rendition> elements, each of which must bear a unique identifier, followed by a sequence of <tagUsage> elements, one for each distinct element occurring within the outermost <text> element of a TEI document.

The <rendition> element defined in this version of the TEI Guidelines is a preliminary proposal only, intended to provide a hook for more detailed specifications of default rendition in later versions.

The present proposal allows the encoder to enter an informal description of a rendition, or style, as running prose only. This rendition will be assumed to apply, by default, to all occurrences of an element which names its identifier as the value of the render attribute of the appropriate <tagUsage> element.

For element occurrences to which this default rendition does not apply, the encoder should specify an explicit description using the global rend attribute on the elements concerned.

For example, the following schematic shows how an encoder might specify that <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 id="style1">
      ... description of one default rendition here ...
   </rendition>
   <rendition id="style2">
      ... description of another default rendition here ...
   </rendition>
   <tagUsage gi="p" render="style1"> ... </tagUsage>
   <tagUsage gi="hi" render="style2"> ... </tagUsage>
   <!-- ... -->
</tagsDecl>
```

No detailed proposals for the content of the <rendition> element have as yet been formulated. Earlier versions of these Guidelines suggested that specifications derived from, or compatible with, the properties standardized as part of the Document Style and Semantics Specification Language (ISO/IEC 10179) might be useful; the Cascading Stylesheet Language (http://www.w3.org/TR/REC-CSS1) is another possible candidate vehicle for their expression, as is the XML vocabulary for specifying formatting semantics which forms a part of the W3C's Extensible Stylesheet Language (http://www.w3.org/TR/xsl).

A <tagsDecl> need not specify any <rendition> element. It must however contain exactly one occurrence of a <tagUsage> element for each distinct element marked within the outermost <text> element associated with the <teiHeader> in which it appears.[67] 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>
```

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 phrases only.

The ident attribute may optionally be used to specify how many of the occurrences of the element in question bear a value for the global id attribute, as in the following example:

```
<tagUsage gi="pb" occurs="321" ident="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 contain a <tagsDecl> element, but if one is present, it must contain <tagUsage> elements for each distinct element marked in the associated text, and the counts specified by their usage attributes must correspond with the number of such elements present in the document, as identified by some conforming processor.

```
<!-- 5.3.4: Tag usage and rendition declarations-->
<!ELEMENT tagsDecl %om.RO; (rendition*, tagUsage*) >
<!ATTLIST tagsDecl
      %a.global;
```

[67] In the case of a TEI corpus (23 *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.

```
      TEIform CDATA 'tagsDecl'  >
<!ELEMENT tagUsage %om.RO; %paraContent; >
<!ATTLIST tagUsage
      %a.global;
      gi CDATA #REQUIRED
      occurs CDATA #IMPLIED
      ident CDATA #IMPLIED
      render IDREF #IMPLIED
      TEIform CDATA 'tagUsage'  >
<!ELEMENT rendition %om.RO; %paraContent; >
<!ATTLIST rendition
      %a.global;
      TEIform CDATA 'rendition'  >
<!-- end of 5.3.4-->
```

5.3.5 The Reference System Declaration

The <refsDecl> element is the fifth of the nine optional subdivisions of the <encodingDesc> element. It is used to document the way in which any standard referencing scheme built into the encoding works, either as a series of prose paragraphs or by using the following specialized elements:

<refsDecl> specifies how canonical references are constructed for this text. Attributes include:

doctype identifies the *document type* within which this reference declaration is used.

Values must be the name of a document type

<step> specifies one component of a canonical reference defined by the "stepwise" method. Attributes include:

refunit (reference unit) names the unit (book, chapter, canto, verse, ...) identified by this step in a canonical reference.

Values any string of characters; typically a word or phrase in some natural language.

length specifies the fixed length of the reference component.

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.

delim supplies a delimiting string following the reference component.

Values If a single space is used it is interpreted as whitespace

from specifies the starting point of the area referred to by this step in the canonical reference.

Values a valid expression in the TEI extended pointer notation documented in section 14.2 *Extended Pointers*.

to specifies the ending point of the area referred to by this step in the canonical reference.

Values a valid expression in the TEI extended pointer notation documented in section 14.2 *Extended Pointers*.

<state> specifies one component of a canonical reference defined by the "milestone" method. Attributes include:

ed (edition) indicates which edition or version the milestone applies to.

Values Any string of characters; usually a siglum conventionally used for the edition.

unit indicates what kind of state is changing at this milestone.

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.

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.

delim supplies a delimiting string following the reference component.

Values If a single space is used it is interpreted as whitespace.

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 6.9 *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 *steps* expressed in the TEI extended pointer notation (documented in section 14.2 *Extended Pointers*)
- 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. A reference declaration can only describe the referencing system applicable to a single document type; if therefore concurrent document types are in use (as discussed in section 6.9 *Reference Systems*), a `<refsDecl>` element must be supplied for each; the doctype attribute should be used to specify the document type to which the declaration relates.

5.3.5.1 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 N 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 HEAD element within each text. The N
    attribute on each DIV1 and DIV2 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 LINEBREAK elements, each of which
    includes the through line number in Casaubon's edition as the
    value of its N 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 N values from the TEXT, DIV1, or DIV2 and calculating the line
    number within each part.</p>
</refsDecl>
```

5.3.5.2 Stepwise Method

This method defines each reference as a series of *steps*, each of which corresponds to a single pair of expressions in the TEI extended pointer notation (for which see section 14.2 *Extended Pointers*). Often, but not always, each step will also correspond to one portion of the canonical reference itself; in many common forms of canonical reference, each step will narrow the scope within which the next step can be taken. The `<refsDecl>` element must specify the steps, delimiters, and lengths to be used by an application program, both when constructing references for a given location and when interpreting canonical references within a given document hierarchy. It does so by supplying one or more `<step>`

elements, each of which identifies the type of 'reference unit' handled by the step and uses a pair of extended-pointer expressions to indicate the starting and ending pointers of the portion of the document which corresponds to a given portion of the reference string. The element may also give either a delimiter or a length for use in breaking the corresponding reference string up into units.

<step> specifies one component of a canonical reference defined by the "stepwise" method. Attributes include:

refunit (reference unit) names the unit (book, chapter, canto, verse, ...) identified by this step in a canonical reference.
Values any string of characters; typically a word or phrase in some natural language.

from specifies the starting point of the area referred to by this step in the canonical reference.
Values a valid expression in the TEI extended pointer notation documented in section 14.2 *Extended Pointers*.

to specifies the ending point of the area referred to by this step in the canonical reference.
Values a valid expression in the TEI extended pointer notation documented in section 14.2 *Extended Pointers*.

delim supplies a delimiting string following the reference component.
Values If a single space is used it is interpreted as whitespace

length specifies the fixed length of the reference component.
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.

For example, the reference "Matthew 5:29" might be constructed by stepping down the tree to find an element labelled as the "Matthew" node, then within that to the "5" node, and finally, within that, to the "29" node. The following declarations would be required; the special values %1, %2, and %3 refer here to the strings 'Matthew', '5', and '29', respectively.

```
<refsDecl>
  <step refunit="book"    delim=" " from="DESCENDANT (1 DIV1 N %1)"/>
  <step refunit="chapter" delim=":" from="DESCENDANT (1 DIV2 N %2)"/>
  <step refunit="verse"             from="DESCENDANT (1 DIV3 N %3)"/>
</refsDecl>
```

As this example also shows, the steps of such a reference are typically separated by fixed character sequences, called *delimiters*. In this example, the delimiters are a space (following "Matthew") and a colon (following the chapter number). A processor for canonical references would use the delimiters specified by the delim attributes to break the reference string up into pieces; the pieces would then be used to interpret the %1, etc., in the extended pointer expressions of the from and to attributes.

An alternative to the use of delimiters is to specify a fixed length for each step of the reference: for example, the same reference might be given as "MAT05029", assuming a fixed length of 3 for the first step, 2 for the second, and 3 for the third.

```
<refsDecl>
  <step length="3" from="DESCENDANT (1 DIV1 N %1)"/>
  <step length="2" from="DESCENDANT (1 DIV2 N %2)"/>
  <step length="3" from="DESCENDANT (1 DIV3 N %3)"/>
</refsDecl>
```

The order in which the <step> elements are supplied corresponds here with the order of elements within the reference, with the largest (that is, the one nearest the top of the document hierarchy) item first and the smallest last.

For a description of the processing required when a canonical reference defined by <step> elements is to be recognized, and examples of its use, see chapter 32 *Algorithm for Recognizing Canonical References*.

5.3.5.3 Milestone Method

This method is appropriate when only 'milestone' tags (see section 6.9.3 *Milestone Tags*) are available to provide the required referencing information. It does not provide any abilities which cannot be mimicked by the stepwise 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 *state* of a variable. A reference declaration using this method therefore specifies the individual components of the canonical reference as a sequence of `<state>` elements:

<state> specifies one component of a canonical reference defined by the "milestone" method. Attributes include:

ed (edition) indicates which edition or version the milestone applies to.
Values Any string of characters; usually a siglum conventionally used for the edition.

unit indicates what kind of state is changing at this milestone.
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.

delim supplies a delimiting string following the reference component.
Values If a single space is used it is interpreted as whitespace.

length specifies the fixed length of the reference component.
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.

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.[68] 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 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 `<state>` 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>
   <state ed="first" unit="page" length="2" delim="."/>
```

[68] On the milestone tag itself, what are here referred to as 'variables' are identified by the combination of the `ed` and `unit` attributes.

```
   <state ed="first" unit="line" length="4"/>
</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 6.9.3 *Milestone Tags*.

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.

The elements discussed in this section are formally defined as follows:

```
<!-- 5.3.5.3: The reference scheme declaration-->
<!ELEMENT refsDecl %om.RO; (p+ | step+ | state+)>
<!ATTLIST refsDecl
      %a.global;
      doctype CDATA "TEI.2"
      TEIform CDATA 'refsDecl'  >
<!ELEMENT step %om.RO; EMPTY>
<!ATTLIST step
      %a.global;
      refunit CDATA #IMPLIED
      length CDATA #IMPLIED
      delim CDATA #IMPLIED
      from %extPtr; #REQUIRED
      to %extPtr; "DITTO"
      TEIform CDATA 'step'  >
<!ELEMENT state %om.RO; EMPTY>
<!ATTLIST state
      %a.global;
      ed CDATA #IMPLIED
      unit CDATA #REQUIRED
      length CDATA #IMPLIED
      delim CDATA #IMPLIED
      TEIform CDATA 'state'  >
<!-- end of 5.3.5.3-->
```

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 23.3 *Associating Contextual Information with a Text*.

5.3.6 The Classification Declaration

The <classDecl> element is the sixth of the nine optional subdivisions of the <encodingDesc> element. It 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> 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> 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 id="ddc12">
  <bibl>
    <title>Dewey Decimal Classification</title>
    <edition>Abridged Edition 12</edition>
    <!-- etc. -->
  </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 a <textDesc> element defining it in terms of the situational parameters discussed in section 23.2.1 *The Text Description*. 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 id attribute, which is used as the target for <catRef> elements referring to it.

```
<taxonomy id="b">
  <bibl>Brown Corpus</bibl>
  <category id="b.a">
    <catDesc>Press Reportage</catDesc>
    <category id="b.a1"><catDesc>Daily</catDesc></category>
    <category id="b.a2"><catDesc>Sunday</catDesc></category>
    <category id="b.a3"><catDesc>National</catDesc></category>
    <category id="b.a4"><catDesc>Provincial</catDesc></category>
    <category id="b.a5"><catDesc>Political</catDesc></category>
    <category id="b.a6"><catDesc>Sports</catDesc></category>
    <!-- ... -->
  </category>
  <category id="b.d"><catDesc>Religion</catDesc>
    <category id="b.d1"><catDesc>Books</catDesc></category>
    <category 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 5.4.3 *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"/>
```

The elements discussed in this section are defined as follows:

```
<!-- 5.3.6: The classification declaration-->
<!ELEMENT classDecl %om.RR; (taxonomy+)>
<!ATTLIST classDecl
      %a.global;
      TEIform CDATA 'classDecl'  >
<!ELEMENT taxonomy %om.RR; (category+ | ((bibl | biblStruct | biblFull),
                                  category*))>
<!ATTLIST taxonomy
      %a.global;
      TEIform CDATA 'taxonomy'  >
<!ELEMENT category %om.RR; (catDesc, category*)>
<!ATTLIST category
```

```
      %a.global;
      TEIform CDATA 'category'  >
<!ELEMENT catDesc %om.RO; (#PCDATA | %m.phrase; | textDesc )*>
<!ATTLIST catDesc
      %a.global;
      TEIform CDATA 'catDesc'  >
<!-- end of 5.3.6-->
```

5.3.7 The Feature System Declaration

The <fsdDecl> element is the seventh of the nine optional subdivisions of the <encodingDesc> element. It is used to associate a *feature system declaration* (as defined in chapter 26 *Feature System Declaration*) with any analytic *feature structures* (as defined in chapter 16 *Feature Structures*) present in the text documented by this header.

It has the following description and attributes:

<fsdDecl> identifies the feature system declaration which contains definitions for a particular type of feature structure. Attributes include:

type identifies the type of feature structure documented in the FSD; this will be the value of the type attribute on at least one feature structure.
Values any string of characters.

fsd (feature-system declaration) specifies the external entity containing the feature system declaration; an entity declaration in the document's DTD subset must associate the entity name with a file on the system.
Values a valid external entity name

Note that one <fsdDecl> element must be specified for each distinct type of feature structure used in the markup. The fsd element supplies the name of an external entity containing a feature system declaration in which that type of feature structure is defined (see further chapter 26 *Feature System Declaration*). This external entity does not use the same DTD as the rest of the document; it is recommended therefore to declare it as an unparsed external entity using a foreign notation. The following document type subset for the document shows how this may be achieved in either XML or SGML.

```
<!DOCTYPE TEI.2 PUBLIC "-//TEI P4//DTD Main Document Type//EN" "tei2.dtd" [
 <!-- Declare the fsd notation itself -->
 <!NOTATION fsd
          PUBLIC "-//TEI//Feature System Declaration (1994)//EN">
 <!-- Declare the external entity containing the FSD -->
  <!ENTITY myFeatures  SYSTEM 'myfeat.fsd' NDATA FSD >
]>
<TEI.2>
<!-- ... -->
</TEI.2>
```

This declaration associate the name myFeatures with an external unparsed entity located by the SYSTEM identifier myfeat.fsd, which uses the FSD notation, itself declared above and associated with an appropriate PUBLIC identifier. [69] This entity name may then be specified within an <fsdDecl> element in the header to inform a processor of the location of the feature system declaration corresponding to a given type of feature structure used within the text, as follows:

```
<teiHeader>
   <fileDesc> <!-- ... --> </fileDesc>
   <encodingDesc>
      <!-- ... -->
      <fsdDecl type='myA1' fsd='myFeatures'/>
      <fsdDecl type='myA2' fsd='myFeatures'/>
      <!-- ... -->
   </encodingDesc>
   <!-- ... -->
</teiHeader>
```

[69] In an SGML context, the external entity might alternatively be declared using the SUBDOC keyword to indicate that this entity contains SGML data which can be parsed using some other DTD than the current one. Since SUBDOC entities are not provided in XML, this is not recommended for general usage.

This header would be attached to a text in which feature structures of types myA1 and myA2 are used.

Further details and examples of the use of feature structure analyses and feature system declarations are provided in chapters 16 *Feature Structures* and 26 *Feature System Declaration* respectively.

The <fsdDecl> element is declared as follows:

```
<!-- 5.3.7: The FSD declaration-->
<!ELEMENT fsdDecl %om.RO;  EMPTY>
<!ATTLIST fsdDecl
      %a.global;
      type CDATA #REQUIRED
      fsd ENTITY #REQUIRED
      TEIform CDATA 'fsdDecl'  >
<!-- end of 5.3.7-->
```

5.3.8 The Metrical Declaration Element

The <metDecl> element is the eighth of the nine optional subdivisions of the <encodingDesc> element. It is used to document any metrical notation scheme used in the text, as further discussed in section 9.4 *Rhyme and Metrical Analysis*. It consists either of a prose description or a series of <symbol> elements.

<metDecl> 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>). Attributes include:

pattern specifies a regular expression defining any value that is legal for this notation.

Values the value must be a valid expression for the PATTERN keyword as defined in the TEI extended pointer notation (see section 14.2.2.14 *The PATTERN Keyword*).

<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 <metNotation>. Attributes include:

value specifies the character or character sequence being documented.

Values any available character or character sequence.

terminal specifies whether the symbol is defined in terms of other symbols (terminal="N") or in prose (terminal="Y").

Legal values are:

- Y the element contains a prose definition of its meaning.
- N the element contains a definition of its meaning given using symbols defined elsewhere in the same metNotation element.

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 <symbol> 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 <symbol> 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, foot 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)+\|?/?)*">
   <symbol value="1">metrical promimence</symbol>
   <symbol value="0">metrical non-prominence</symbol>
   <symbol value="|">foot boundary</symbol>
   <symbol value="/">metrical line boundary</symbol>
</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]+">
   <symbol n="dactyl" value="D" terminal="N">-oo</symbol>
   <symbol n="trochee" value="T" terminal="N">-o</symbol>
   <symbol n="iamb" value="I" terminal="N">o-</symbol>
   <symbol n="spondee" value="S" terminal="N">--</symbol>
   <symbol n="tribrach" value="3" terminal="N">ooo</symbol>
   <symbol n="anapaest" value="A" terminal="N">oo-</symbol>
   <symbol value="o">short syllable</symbol>
   <symbol value="-">long syllable</symbol>
</metDecl>
```

Note here the use of the global n attribute to supply an additional name for the symbols being documented.

For further discussion of this metrical notation and its use in the encoding of verse, see section 9.4 *Rhyme and Metrical Analysis*.

The elements discussed in this section are defined as follows:

```
<!-- 5.3.8: Metrical Notation Declaration-->
<!ELEMENT metDecl %om.RO;  ((%component.seq;) | (symbol+))>
<!ATTLIST metDecl
      %a.global;
      %a.declarable;
      type CDATA "met real"
      pattern CDATA #IMPLIED
      TEIform CDATA 'metDecl'  >
<!ELEMENT symbol %om.RO;  %phrase.seq;>
<!ATTLIST symbol
      %a.global;
      value CDATA #REQUIRED
      terminal ( Y | N ) "Y"
      TEIform CDATA 'symbol'  >
<!-- end of 5.3.8-->
```

5.3.9 The Variant-Encoding Method Element

The <variantEncoding> element is the last of the nine optional subdivisions of the <encodingDesc> element. It is used to document the method used to encode textual variants in the text, as discussed in section 19.2 *Linking the Apparatus to the Text*.

<variantEncoding> declares the method used to encode text-critical variants. Attributes include:

method indicates which method is used to encode the apparatus of variants.

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.

location indicates whether the apparatus appears within the running text or external to it.

Legal values are:

`internal` apparatus appears within the running text.

`external` apparatus appears outside the base text.

Its formal declaration is as follows:

```
<!-- 5.3.9: Variant-Encoding Declaration-->
<!ELEMENT variantEncoding %om.RO;  EMPTY>
<!ATTLIST variantEncoding
      %a.global;
      method (location-referenced | double-end-point |
parallel-segmentation) #REQUIRED
      location (internal | external) #REQUIRED
      TEIform CDATA 'variantEncoding'  >
<!-- end of 5.3.9-->
```

5.4 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> 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> describes the languages, sublanguages, registers, dialects etc. represented within a text.

<textClass> 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 additional tag set for the TEI header is in use:

<textDesc> provides a description of a text in terms of its *situational parameters*.

<particDesc> describes the identifiable speakers, voices, or other participants in a linguistic interaction.

<settingDesc> 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 23.2 *Contextual Information*.

Finally, the following element can appear in the <profileDesc> element, when the additional tag set for transcription of primary sources is selected:

<handList> contains a series of <hand> elements listing the different hands of the source.

For a description of this element, see section 18.2.1 *Document Hands*.

The profile description itself has the following formal definition:

```
<!-- 5.4: The profile description-->
<!ELEMENT profileDesc %om.RR;  (creation?, langUsage*,
textDesc*,particDesc*, settingDesc*, handList*, textClass*)>
<!ATTLIST profileDesc
      %a.global;
      TEIform CDATA 'profileDesc'  >
<!--declarations from 5.4.1: Creation inserted here -->
<!--declarations from 5.4.2: Language usage inserted here -->
<!--declarations from 5.4.3: Text Classification inserted here -->
<!-- end of 5.4-->
```

5.4.1 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 value="1992-08">August 1992</date>
   <rs type="city">Taos, New Mexico</rs>
</creation>
```

The formal declaration of <creation> is as follows:

```
<!-- 5.4.1: Creation-->
<!ELEMENT creation %om.RO; %phrase.seq;>
<!ATTLIST creation
      %a.global;
      TEIform CDATA 'creation'  >
<!-- end of 5.4.1-->
```

5.4.2 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 takes attributes specifying the *writing system* used (see section 4 *Languages and Character Sets*) and the quantity of that language present in the text. Following the <language> elements, prose description may also be added to specify further relevant information.

<langUsage> describes the languages, sublanguages, registers, dialects etc. represented within a text.

<language> characterizes a single language or sublanguage used within a text. Attributes include:

- **wsd** specifies the entity containing the *writing system declaration* used for representing texts in this language.
  - *Values* the named entity should contain a full writing system declaration conforming to the auxiliary WSD document type declaration.
- **usage** specifies the approximate percentage (by volume) of the text which uses this language.
  - *Values* a whole number between 1 and 100

Each <language> element links the document to the formal writing system declaration defining that language and its script; for that reason, its use is recommended. The wsd attribute must give the name of an entity containing a writing system declaration; typically, this will be an external file declared in the document type declaration. For examples and discussion, see 25.6 *Linkage between WSD and Main Document*.

When two sublanguages share the same language code and writing system declaration but are distinguished in the <langUsage> element, only one of the <language> elements should bear the id attribute:

```
<langUsage>
   <language id="fr" wsd="wsd.fr" usage="60">Qu&eacute;becois</language>
   <language id="en" wsd="wsd.en" usage="20">Canadian business English</language>
   <language wsd="wsd.en" usage="20">British English</language>
</langUsage>
```

or, less formally,

```
<langUsage>
  <language id="fr" wsd="wsd.fr">   </language>
  <language id="en" wsd="wsd.en">
    Approximately two-thirds of the text is in
    Qu&eacute;becois, the remainder being equally
    divided between Canadian business English and
    British English.
  </language>
</langUsage>
```

The <langUsage> and <language> elements have the following formal definitions:

```
<!-- 5.4.2: Language usage-->
<!ELEMENT langUsage %om.RO; (p | language)+>
<!ATTLIST langUsage
      %a.global;
      %a.declarable;
      TEIform CDATA 'langUsage'  >
<!ELEMENT language %om.RO; %phrase.seq;>
<!ATTLIST language
      %a.global;
      wsd ENTITY #IMPLIED
      usage CDATA #IMPLIED
      TEIform CDATA 'language'  >
<!-- end of 5.4.2-->
```

5.4.3 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 23.2.1 *The Text Description*, or the demographic elements described in section 23.2.2 *The Participants 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 tags are provided for this purpose:

<keywords> contains a list of keywords or phrases identifying the topic or nature of a text. Attributes include:

scheme identifies the controlled vocabulary within which the set of keywords concerned is defined.
Values identifier of the associated <taxonomy> element

<classCode> contains the classification code used for this text in some standard classification system. Attributes include:
scheme identifies the classification system or taxonomy in use.
Values must identify a <taxonomy> element.

<catRef> specifies one or more defined categories within some taxonomy or text typology. Attributes include:
target identifies the categories concerned
Values One or more identifiers for <category> elements defined in the current document.

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 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 5.3.6 *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, in the same way as for the <keywords> element, as in the following example:

```
<classCode scheme="ddc19">005.756</classCode>
<classCode scheme="lc">QA76.9</classCode>

<classCode scheme="ddc19">820.285</classCode>
<classCode scheme="lc">PR21</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 5.3.6 *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="b1 b2 b5"/>
```

Where more than one descriptive taxonomy is used to characterize the texts in a corpus or collection, the scheme attribute should be supplied to specify the taxonomy to which the categories identified by the target attribute belong. For example,

```
<catRef target="b12 b15" scheme="brown"/>
<catRef target="a45" scheme="suc"/>
```

Here the same text has been classified as of categories “B12” and “B15” within the Brown classification scheme, and as of category “A45” within the SUC classification scheme.

The distinction between the <catRef> and <classCode> elements is that the values used as identifying codes *must* be defined somewhere within the header for the former, but not the latter.

The elements described in this section have the following formal definitions:

```
<!-- 5.4.3: Text Classification-->
<!ELEMENT textClass %om.RR; ((classCode | catRef | keywords)* )>
<!ATTLIST textClass
      %a.global;
      %a.declarable;
      TEIform CDATA 'textClass'  >
<!ELEMENT keywords %om.RO; (term+ | list)>
<!ATTLIST keywords
      %a.global;
      scheme IDREF #IMPLIED
      TEIform CDATA 'keywords'  >
<!ELEMENT classCode %om.RR; %phrase.seq;>
<!ATTLIST classCode
      %a.global;
      scheme IDREF #IMPLIED
      TEIform CDATA 'classCode'  >
<!ELEMENT catRef %om.RO; EMPTY>
<!ATTLIST catRef
      %a.global;
      target IDREFS #REQUIRED
      scheme IDREF #IMPLIED
      TEIform CDATA 'catRef'  >
<!-- end of 5.4.3-->
```

5.5 The Revision Description

The final subelement 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> 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 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 6.7 *Lists* or as a series of special purpose <change> elements, each of which has the following constituents:

<date> contains a date in any format. Attributes include:

value gives the value of the date in some standard form, usually yyyy-mm-dd.

Values Any string representing a date in standard format; recommended form is ISO 8601:2000 5.2.1.1 Complete representation, extended format (yyyy-mm-dd)

certainty indicates the degree of precision to be attributed to the date.

Values Any appropriate value, e.g. *ca.*, *approx*, *after*, *before*.

<respStmt> supplies a statement of responsibility for someone responsible 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.

<item> contains one component of a list.

The <date> element indicates the date of the change. The <respStmt> element indicates who made the change, and in what role. The <item> element indicates what change was made; it 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), use the global n attribute on the <change> element to supply it.

It is recommended to give changes in reverse chronological order, most recent first.

For example:

```
<revisionDesc>
<change n="RCS:1.70"><date value="2001-04-11">Wed, 11 Apr 01</date>
    <respStmt><name key="zmizuho.zgk">Zo&euml; Mizuho</name>
    <resp>encoder</resp></respStmt>
    <item>made correx entries (unfinished from <date
        value="1999-03-05">March 1999</date>.) all in text not tag.
      supravalidated.</item>
    </change>
<change n="RCS:1.62"><date value="1999-08-24">Tue, 24 Aug 99</date>
    <respStmt><name key="jrussom.zxg">Jacque Russom</name>
    <resp>encoder</resp></respStmt>
    <item>Removed vuji markup from FOREIGN and BIBL contents;
          standardized vuji tags.</item>
    </change>
<change n="RCS:1.47"><date value="1999-07-05">Mon, 05 Jul 99</date>
    <respStmt><name key="edillon.pal">Erica Dillon</name>
    <resp>encoder</resp></respStmt>
    <item>Deleted cit elements in Concluding matter, and propagated
          attributes from this element to the q element. However,
          only q elements exist in this text, where the quote element
          actually seems to be more appropriate. This should be
          looked into.</item>
    </change>
<change n="RCS:1.45"><date value="1999-06-30">Wed, 30 Jun 99</date>
    <respStmt><name key="cmah.dci">Carole Mah</name>
    <resp>encoder</resp></respStmt>
    <item>Fixed n= attribute on PB; they were mis-numbered.</item>
    </change>
<change n="RCS:1.43"><date value="1999-06-18">Fri, 18 Jun 99</date>
    <respStmt><name key="pcaton.xzc">Paul Caton</name>
    <resp>Electronic Publications Editor</resp></respStmt>
    <item>Within FIGURE, moved P to its correct position after
          FIGDESC. Validated against DTD 1.1.30b</item>
    </change>
<change><date value="1998-07-04">04 July 1998</date>
    <respStmt><name key="kmessman.yec">Kevin Messman</name>
    <resp>Encoder</resp></respStmt>
    <item>Double-proofed and entered final corrections.</item>
    </change>
<change><date value="1997-06-12">12 June 1997</date>
    <respStmt><name key="lmayer.ins">Lauryn S. Mayer</name>
    <resp>Encoder</resp></respStmt>
    <item>Began entering corrections with version 1.1.2a of DTD</item>
    </change>
<change><date value="1997-03-13">13 March 1997</date>
    <respStmt><name key="lmayer.ins">Lauryn S. Mayer</name>
    <resp>Encoder</resp></respStmt>
    <item>Began capture using Author/Editor v. 3.1 on Mac with
          version 1.0.14 of DTD.</item>
    </change>
```

```
</revisionDesc>
```

The formal definition of the <revisionDesc> element is thus as follows:

```
<!-- 5.5: The Revision Description-->
<!ELEMENT revisionDesc %om.RR; (list | change+)>
<!ATTLIST revisionDesc
      %a.global;
      TEIform CDATA 'revisionDesc'  >
<!ELEMENT change %om.RO; (date, respStmt+, item)>
<!ATTLIST change
      %a.global;
      TEIform CDATA 'change'  >
<!--respStmt, item, and date are declared in teicore2.-->
<!-- end of 5.5-->
```

5.6 Minimal and Recommended Headers

The TEI header allows for the provision of a very large amount of information concerning the text itself, its source, encodings and revisions of it, as well as a wealth of descriptive information such as the languages it uses and the situation within which it was produced, 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 printed 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>Standard date values are given in ISO form:
           yyyy-mm-dd. </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 id="lcsh">
            <bibl>Library of Congress Subject Headings</bibl>
         </taxonomy>
         <taxonomy id="lc">
            <bibl>Library of Congress Classification</bibl>
         </taxonomy>
      </classDecl>
   </encodingDesc>
   <profileDesc>
      <creation>
         <date>1774</date>
      </creation>
      <langUsage>
         <language id="en" wsd="english" usage="100">English.</language>
      </langUsage>
      <textClass>
         <keywords scheme="lcsh">
            <list>
               <item>Political science</item>
               <item>United States -- Politics and government &mdash;
                      Revolution, 1775-1783</item>
            </list>
         </keywords>
         <classCode scheme="lc">JC 177</classCode>
      </textClass>
   </profileDesc>
   <revisionDesc>
      <change>
         <date>1996-01-22</date>
         <respStmt>
            <resp>ed</resp>
```

```
            <name>CMSMcQ</name>
         </respStmt>
         <item>finished proofreading</item>
      </change>
      <change>
         <date>1995-10-30</date>
         <respStmt>
            <resp>ed</resp>
            <name>L.B. </name>
         </respStmt>
         <item>finished proofreading</item>
      </change>
      <change>
         <date>1995-07-20</date>
         <respStmt>
            <resp>ed</resp>
            <name>R.G. </name>
         </respStmt>
         <item>finished proofreading</item>
      </change>
      <change>
         <date>1995-07-04</date>
         <respStmt>
            <resp>ed</resp>
            <name>R.G. </name>
         </respStmt>
         <item>finished data entry</item>
      </change>
      <change>
         <date>1995-01-15</date>
         <respStmt>
            <resp>ed</resp>
            <name>R.G. </name>
         </respStmt>
         <item>began data entry</item>
      </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.

5.7 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 the machine-readable data file. 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(CF) for computer files. 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) were published in 1978, with a revision appearing in 1988. The AACR2 provides guidelines for the construction of catalogues in general libraries. 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 catalogue 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.