The Association for Computers and the Humanities (ACH)

The Association for Computational Linguistics (ACL)

The Association for Literary and Linguistic Computing (ALLC)

Guidelines for Electronic Text Encoding and Interchange

Edited by

C. M. Sperberg-McQueen and Lou Burnard

TEI P2, Chapter 9 VE
Text Encoding Initiative
Chicago, Oxford

November 13, 1993

Draft Version 2, November 13, 1993

Note

This is the current draft (as of November 13, 1993) of one section of the *Guidelines for Electronic Text Encoding and Interchange*, version 2 (also known as TEI P2), produced by the Text Encoding Initiative. This fascicle contains:

- a table of contents for TEI P2, indicating the status of each section as of the release date of this fascicle
- a draft for one section of TEI P2
- entries derived from this section for the reference list of tags of TEI P2 (if any)
- a User Response and Comment Form

Please feel free to send us your comments on the contents of this draft and their applicability to your needs. Comments may be sent to the TEI-L list server for general discussion, or directly to either of the TEI editors (adresses on the User Response and Comment form). Bear in mind in reading, however, that TEI P2 is a reference manual, not a tutorial in the use of electronic text, and unavoidably assumes either a certain amount of prior knowledge or patience with unfamiliar technical material. Note also that some matters which may appear obscure in one section may well be treated in more detail elsewhere in the full text of TEI P2.

TEI P2, like P1, is a working draft. The observant will find several typographic and presentational flaws in the current text: for example, cross-references are not properly resolved in the LaTeX versions and some parts of the reference material are not yet complete. We hope to rectify as many as possible of these flaws in the coming months, but have decided not to allow them to delay any further the release of the draft. We hope that those interested in the Guidelines will find the current state of the text readable and usable enough for the purpose of substantive comment. As far as possible, all comments received on this section will be taken into account in the preparation of the next version of the Guidelines, which is due by the end of 1993.

Updates on the status of the draft as a whole will be distributed automatically to all subscribers of the Listserv list TEI-L. To subscribe to TEI-L, send electronic mail to the address LISTSERV@UICVM (or Listserv@uicvm.uic.edu) containing the single line

```
subscribe tei-l J. Smith
```

(substituting your name for “J. Smith”). TEI-L is also the appropriate place to pose questions or offer public comments on the TEI guidelines and other relevant issues. The TEI-L file server contains all the sections of TEI P2 thus far released, as well as other TEI materials. Its contents are shadowed at various other sites around the world; for more information consult one of the following documents:

- TEI ED J8, “Obtaining the Second Version of the TEI Guidelines (TEI P2)” (describes how to retrieve electronic copies of TEI P2 and the various formats they are available in)
- TEI ED J9, “Obtaining Paper Copies of the Second Version of the TEI Guidelines (TEI P2)” (describes how to request paper copies of TEI P2, for those without electronic mail access)

These documents are available from the TEI-L file server or through the editors at the addresses given on the User Response and Comment Form.

Overall Table of Contents for TEI P2

The following list shows the expected contents of TEI P2 as a whole and indicates the current status of each chapter.

Contents

Part 2

Core Tags and General Rules

Chapter 9

Base Tag Set for Verse

This base tag set is intended for use when encoding texts which are entirely or predominantly in verse, and for which the elements for encoding verse structure already provided by the core tag set are inadequate.

The tags described in section 6.11.1, “Core Tags for Verse,” [in separate fascicle] include elements for the encoding of verse lines and line groups such as stanzas: these are available for any TEI document, irrespective of the base tag set it uses. Like the base tag sets for prose and for drama, the base tag set for verse additionally makes use of the tag set defined in chapter 7, “Default Text Structure,” [in separate fascicle] to define the basic formal structure of a text, in terms of **<front>**, **<body>** and **<back>** elements and the text-division elements into which these may be subdivided.

The base tag set for verse extends the facilities provided by these two tag sets in the following ways:

- “numbered” **<lg>** elements are provided, by analogy with the “numbered” *divn* class elements discussed in section 7.1, “Divisions of the Body,” [in separate fascicle] (see section 9.2, “Structural Divisions of Verse Texts,” on page 5)
- a special purpose **<caesura>** element is provided, to allow for segmentation of the verse line (see section 9.3, “Components of the Verse Line,” on page 9)
- a set of attributes is provided for the encoding of rhyme scheme and metrical information (see sections 9.4, “Rhyme and Metrical Analysis,” on page 12 and 9.5, “Rhyme,” on page 18)

To enable the base tag set for verse texts, a parameter entity *TEI.verse* must be declared within the document type subset, the value of which is “INCLUDE”, as further described in section 3.3, “Invocation of TEI DTDs,” [in separate fascicle]. A document using this base tag set and no additional tag sets will thus begin as follows:

```
<!DOCTYPE TEI.2 system 'tei2.dtd' [
     <!ENTITY % TEI.verse 'INCLUDE' >
]>
```

This declaration makes available the elements and attributes described in this chapter, in addition to those described in chapters 6, “Tags Available in All TEI Documents,” [in separate fascicle] and 7, “Default Text Structure,” [in separate fascicle].

9.1 Structure of the Base Tag Set for Verse

The base tag set for verse contains the following entity declarations:

```
<!-- 9.1:  Base Tag Set for Verse: entities                  -->
<!-- Text Encoding Initiative: Guidelines for Electronic      -->
<!-- Text Encoding and Interchange. DRAFT Version 2. 1992-93. -->

<!-- Copyright (c) 1990, 1992, 1993 ACH, ACL, ALLC.           -->
<!-- Permission to copy in any form is granted for use with   -->
```

```
<!-- TEI-aware systems and applications, provided this        -->
<!-- notice is included in all copies.                         -->

<!-- These materials may not be altered; modifications to     -->
<!-- these DTDs should be performed as specified in chapter   -->
<!-- MD of the Guidelines.                                     -->

<!-- These materials subject to revision. Current versions    -->
<!-- available from the Text Encoding Initiative.             -->


<!-- First, declare the class of components specific to verse -->

<!ENTITY % x.comp.verse '' >
<!ENTITY % m.comp.verse '%x.comp.verse lg1' >
<!-- Next define attributes common to metrical elements       -->

<!ENTITY % a.metrical '
          met                CDATA               %INHERITED
          real               CDATA               #IMPLIED
          rhyme              CDATA               #IMPLIED'      >
<!ENTITY % a.enjamb '
          enjamb             CDATA               #IMPLIED'      >
<!-- Now, we define the class of elements which can appear    -->
<!-- only within metrical elements                             -->

<!ENTITY % x.phrase.verse '' >
<!ENTITY % m.phrase.verse '%x.phrase.verse caesura' >
```

The base tag set for verse contains the following element declarations:

```
<!-- 9.1:  Base Tag Set for Verse                              -->
<!-- First, declare the default text structure, which is the  -->
<!-- same for verse as it is for other kinds of text.          -->

<!ENTITY % tei.structure.dtd system 'teistr2.dtd' >
%tei.structure.dtd;
<!-- Finally, declare the elements specific to this base tag  -->
<!-- set                                                       -->

<!-- ... declarations from section 9.2                         -->
<!--     (Numbered line groups)                                -->
<!--     go here ...                                           -->
<!-- ... declarations from section 9.3                         -->
<!--     (Caesura)                                             -->
<!--     go here ...                                           -->
```

9.2 Structural Divisions of Verse Texts

Like other kinds of text, texts written in verse may be of widely differing lengths and structures. A complete poem, no matter how short, may be treated as a free-standing text, and encoded in the same way as a distinct prose text. A group of poems functioning as a single unit may be encoded either as a **<group>** or as a **<text>**, depending on the encoder's view of the text. For further discussion, including an example encoding for a verse anthology, see chapter 7, “Default Text Structure,” [in separate fascicle].

Many poems consist only of ungrouped lines. This short poem by Emily Dickinson is a simple case:

```
<text>
<front><head>1755</head></front>
<body>
<l>To make a prairie it takes a clover and one bee,</l>
<l>One clover, and a bee,</l>
<l>And revery.</l>
<l>The revery alone will do,</l>
<l>If bees are few.</l>
</text>
```

Often, however, lines are grouped, formally or informally, into stanzas, verse paragraphs, etc. The **<lg>** element defined in the core tag set (in section 6.11.1, “Core Tags for Verse,” [in separate fascicle]) may be used for all such groupings. It may thus serve for informal groupings of lines such as those of the following example from Allen Ginsberg:

```
<text><body><head>My Alba</head>

<lg type=free>
<l>Now that I've wasted</l>
<l>five years in Manhattan</l>
<l>life decaying</l>
<l>talent a blank</l>
</lg>
<lg>
<l>talking disconnected</l>
<l>patient and mental</l>
<l>sliderule and number</l>
<l>machine on a desk</l>
</lg>
<!-- ... -->
</text>
```

It may also be used to mark the verse paragraphs into which longer poems are often divided, as in the following example from Samuel Taylor Coleridge's “Frost at Midnight”:

```
<lg type=para>
<l>The Frost performs its secret ministry,
<l>Unhelped by any wind....
<!-- ... -->
<l>Whose puny flaps and freaks the idling Spirit
<l>By its own moods interprets, every where
<l>Echo or mirror seeking of itself,
<l part=i>And makes a toy of Thought.
</lg><lg>
<l part=f>But O! how oft,
<l>How oft, at school, with most believing mind
<l>Presageful, have I gazed upon the bars,
<l>To watch that fluttering <hi>stranger</hi>! ...
<!-- ... -->
</lg>
<lg>
<l>Dear Babe, that sleepest cradled by my side,
<!-- ... -->
```

Note, in the above example, the use of the *part* attribute on the **<l>** element, where a verse line is broken between two line groups, as discussed in section 6.11.1, “Core Tags for Verse,” [in separate fascicle]. (Note also that here and in some other examples in this chapter the end-tags for the **<l>** element, being redundant, have been omitted, as is allowed when the SGML feature OMITTAG is enabled.)

Most typically, however, the **<lg>** element is used to mark the highly regular line groups which characterize stanzaic and similar verse forms, as in the following example from Chaucer:

```
<lg>
<l>Sire Thopas was a doghty swayn;
<l>White was his face as payndemayn,
<l>His lippes rede as rose;
<l>His rode is lyk scarlet in grayn,
<l>And I yow telle in good certayn,
<l>He hadde a semely nose.
</lg><lg>
<l>His heer, his ber was lyk saffroun,
<l>That to his girdel raughte adoun;
<!-- ... -->
</lg>
```

Like other text-division elements, **<lg>** elements may be nested hierarchically. For example, one particularly common English stanzaic form consists of a quatrain or sestet followed by a couplet. The **<lg>** element may be used to encode both the stanza and its components, as in the following example from Byron:

```
<lg type=stanza>
   <lg type=sestet>
     <l>In the first year of Freedom's second dawn
     <l>Died George the Third; although no tyrant, one
     <l>Who shielded tyrants, till each sense withdrawn
     <l>Left him nor mental nor external sun:
```

```
      <l>A better farmer ne'er brushed dew from lawn,
      <l>A worse king never left a realm undone!
    </lg>
    <lg type=couplet>
      <l>He died &dash; but left his subjects still behind,
      <l>One half as mad &dash; and t'other no less blind.
    </lg>
</lg>
```

Note the use of the *type* attribute to name the type of unit encoded by the **<lg>** element; this attribute is common to all members of the *divn* class (see section 7.1.1, “Un-numbered Divisions,” [in separate fascicle]).[1] “Sestet” and “couplet” might conceivably also be used as the values of the *met* attribute in a metrical analysis, for which see below, section 9.4, “Rhyme and Metrical Analysis,” on page 12. The *type* attribute is intended solely for conventional names of different classes of text block; the *met* attribute is intended for systematic metrical analysis.

The above example uses “un-numbered” line groups which can nest within each other to any depth. When the base tag set for verse is in use, “numbered” line groups may also be used as an alternative.

- **<lg1>** : contains a first-level (i.e. largest) group of verse lines functioning as a formal unit e.g. a stanza, refrain, verse paragraph, etc.
- **<lg2>** : contains a second-level (i.e. second largest) group of verse lines functioning as a formal unit e.g. a stanza, refrain, verse paragraph, etc.

The base tag set for verse defines up to five such numbered line group elements, **<lg1>** to **<lg5>** inclusive. These function in exactly the same way as the numbered *divn* class elements discussed in section 7.1, “Divisions of the Body,” [in separate fascicle].

As an example of their use, consider the Shakespearean sonnet. This may be divided into two parts: a concluding couplet, and a body of twelve lines, itself subdivided into three quatrains:

```
<text><body>
<lg1 type=body>
<lg2 type=quatrain>
<l>My Mistres eyes are nothing like the Sunne,
<l>Currall is farre more red, then her lips red
<l>If snow be white, why then her brests are dun:
<l>If haires be wiers, black wiers grown on her head:
</lg2>
<lg2>
<l>I have seene Roses damaskt, red and white,
<l>But no such Roses see I in her cheekes,
<l>And in some perfumes is there more delight,
<l>Then in the breath that from my Mistres reekes.
</lg2>
```

&@fn@sep

[1] For discussion of other attributes of this class, see 7.1.4, “Partial and Composite Divisions,” [in separate fascicle].

```
<lg2>
<l>I love to heare her speake, yet well I know,
<l>That Musicke hath a farre more pleasing sound:
<l>I graunt I never saw a goddesse goe,
<l>My Mistres when shee walkes treads on the ground.
</lg2>
</lg1>
<lg1 type=couplet>
<l>And yet by heaven I think my love as rare,
<l>As any she beli'd with false compare.
</lg1>
</text>
```

Particularly lengthy poetic texts are often subdivided into units larger than stanzas or paragraphs, which may themselves be subdivided. Spenser's *Faery Queene*, for example, consists of twelve "books" each of which contains a prologue followed twelve "cantos". Each prologue and each canto consists of nine-line "stanzas", each of which follows the same regular pattern. Other examples in the same tradition are easy to find.

Large structures of this kind are most conveniently represented by the *divn* class elements such as **<div>** or **<div1>** described in section 7.1, "Divisions of the Body," [in separate fascicle]. Thus the start of the *Faery Queene* might be encoded as follows:

```
<body>
<div1 type=book n='I'>
  <div2 type=canto n='1'>
     <lg type=stanza n='I.1.1'>
       <l>A noble knight was pricking on the plain
       <l>Ycladd in mightie armes and silver shielde,
  <!-- ... -->
</body>
```

The encoder must choose at which point in the hierarchy of structural units to introduce **<lg>** elements rather than a yet smaller **<div>** element: it would (for example) also be possible to encode the above example as follows:

```
<body>
<div1 type=book n=I>
  <div2 type=canto n=I.1>
     <div3 type=stanza n=I.1.1>
       <l>A noble knight was pricking on the plain
       <l>Ycladd in mightie armes and silver shielde,
       <!-- ... -->
</body>
```

One reason for preferring the former version is that not all of Spenser's stanzaic verse is organized into both cantos and books. In a corpus containing other works as well as the *Faery Queene*, it would be inconvenient for stanzas to appear in one part as **<div3>** elements, and in another as **<div2>** elements.[2]

The numbered line group elements have the following formal definition:

```
<!-- 9.2:  Numbered line groups                                     -->
<!ELEMENT lg1          - o  (head?, (l | lg2)+)                      >
<!ATTLIST lg1               %a.global;
                            %a.divn;
                            %a.metrical;                             >
<!ELEMENT lg2          - o  (head?, (l | lg3)+)                      >
<!ATTLIST lg2               %a.global;
                            %a.divn;
                            %a.metrical;                             >
<!ELEMENT lg3          - o  (head?, (l | lg4)+)                      >
<!ATTLIST lg3               %a.global;
                            %a.divn;
                            %a.metrical;                             >
<!ELEMENT lg4          - o  (head?, (l | lg5)+)                      >
<!ATTLIST lg4               %a.global;
                            %a.divn;
                            %a.metrical;                             >
<!ELEMENT lg5          - o  (head?, l+)                              >
<!ATTLIST lg5               %a.global;
                            %a.divn;
                            %a.metrical;                             >
<!-- This fragment is used in sec. 9.1                              -->
```

9.3 Components of the Verse Line

It is often convenient for various kinds of analysis to encode subdivisions of verse lines. The general purpose **<seg>** element defined in the tag set for segmentation and alignment (section 16.3, “Segmentation,” [in separate fascicle]) is provided for this purpose:

<seg> : contains any arbitrary phrase-level unit of text. Attributes include:
- *type* : indicates what sort of segment is marked.
- *part* : specifies whether or not the segment is complete. Legal values are:
  - *Y* : the segment is incomplete
  - *N* : either the segment is complete, or no claim is made as to its completeness
  - *I* : the initial part of an incomplete segment
  - *M* : a medial part of an incomplete segment
  - *F* : the final part of an incomplete segment

In Old and Middle English alliterative verse, individual verse lines are typically split into half lines. The **<seg>** element may be used to mark these explicitly, as in the following example from Langland's *Piers Plowman*:

[2] Another way of avoiding this problem would be to use un-numbered **<div>** elements; see further 7.1, “Divisions of the Body,” [in separate fascicle].

```
<l><seg>In a somer seson,</seg>
   <seg>whan softe was the sonne,</seg></l>
<l><seg>I shoop me into shroudes</seg>
   <seg>as I a sheep were,</seg></l>
<l><seg>In habite as an heremite </seg>
   <seg>unholy of werkes,</seg></l>
<l><seg>Went wide in this world </seg>
   <seg>wondres to here.</seg></l>
<!-- ... -->
<!-- Langland, Pier Plowman -->
```

The **<seg>** element can be nested hierarchically, in the same way as the **<lg>** element, down to whatever level of detailed structure is required. In the following example, the line has been divided into *feet*, each of which has been further subdivided into into syllables. (The eccentric formatting ensures that no spaces are introduced within words unless they are within SGML tags and thus not significant.)

```
   <l><seg type=foot>
         <seg type=syll>Ar</><seg
              type=syll>ma</>
         <seg type=syll>vi</></seg><seg
      type=foot><seg
              type=syll>rum</><seg
              type=syll>que</>
         <seg type=syll>ca</></seg><seg
      type=foot><seg
              type=syll>no</>
         <seg type=syll>Tro</></seg><seg
      type=foot><seg
              type=syll>iae</>
         <seg type=syll>qui</>
      </seg>
      <seg type=foot>
         <seg type=syll>pri</><seg
              type=syll>mus</>
         <seg type=syll>ab</>
      </seg>
      <seg type=foot>
         <seg type=syll>or</><seg
              type=syll>is</>
      </seg>
</l>
```

The **<seg>** element may be used to identify any subcomponent of a line which has content; its type attribute may characterize such units in any way appropriate to the needs of the encoder. For the specific case of labeling each foot with its formal type ("dactyl", "spondee", etc.), and each syllable with its metrical or prosodic status (syllables bearing primary or secondary stress, long syllables, short syllables), however, the specialized attributes *met* and *real* are defined, which provide a more systematic framework than the *type* attribute; see section 9.4, "Rhyme and Metrical Analysis," on page 12 below.

In classical verse, a hexameter like that above may also be formally divided into two “hemistiches”. This example provides an interesting case, in that the boundary of the first hemistich falls in the middle of one of the feet (between the syllables “no” and “Tro”). If both kinds of segmentation are required, the *part* attribute might be used to mark the overlapping structure as follows. (This example uses the slightly artificial convention that word space is included within the syllable segments, and all other white space is to be ignored; though convenient for visual clarity in presentations like this one, this convention is not recommended for normal use.)

```
<l><seg type=hemistich>
      <seg type=foot>
        <seg type=syll> Ar</>
        <seg type=syll>ma </>
        <seg type=syll>vi</>
      </seg>
      <seg type=foot>
        <seg type=syll>rum</>
        <seg type=syll>que </>
        <seg type=syll>ca</>
        </seg>
      <seg type=foot part=I>
        <seg type=syll>no </>
      </seg>
   </seg>
   <seg type=hemistich>
      <seg type=foot part=F>
        <seg type=syll>Tro</>
      </seg>
      <seg type=foot>
         <seg type=syll>iae </>
         <seg type=syll>qui </>
      </seg>
   <!-- ... -->
   </seg>
```

Numbered **<seg>** elements may also be used; these have the disadvantage—or advantage—of requiring a consistent, layer-by-layer analysis, and the advantage of providing a useful default for the *type* attribute, which must be specified only once for each level, as well as allowing the omission of many end-tags, which makes the presentation more compact.

```
 <l><seg1 type=hemistich>
       <seg2 type=foot>
             <seg3 type=syll> Ar<seg3>ma <seg3>vi</>
       <seg2><seg3>rum<seg3>que <seg3>ca</>
       <seg2 part=I><seg3>no </>
    <seg1>
       <seg2 part=F><seg3>Tro</>
       <seg2><seg3>iae <seg3>qui </>
       <!-- ... -->
  </l>
<!-- ... -->
```

Instead of using the *part* attribute on the **<seg2>** element, it might be simpler just to mark the point at which the caesura occurs. An additional element is provided for analyses of this kind, in which what is to be marked are points “between the words”, which have some significance within a verse line:

<caesura> : marks the point at which a metrical line may be divided.

In classical prosody, the *caesura*, which occurs within a foot, is distinguished from a *diaeresis*, which occurs on a foot boundary (not to be confused with the division of a diphthong into two syllables, or the diacritic symbol used to indicate such division, each of which is also termed *diaeresis*). This distinction is rarely made nowadays, the term *caesura* being used for any division irrespective of foot boundaries. No special-purpose **<diaeresis>** element is therefore provided.

As an example of the **<caesura>** element, we refer again to the example from Langland. An encoder might choose simply to record the location of the caesura within each line, rather than encoding each half-line as a segment in its own right, as follows:

```
<l>In a somer seson, <caesura> whan softe was the sonne, </l>
<l>I shoop me into shroudes <caesura> as I a sheep were, </l>
<l>In habite as an heremite <caesura> unholy of werkes, </l>
<l>Went wide in this world <caesura> wondres to here. </l>
```

Logically, the opposite of caesura might be considered to be enjambement. The base tag set for verse defines an *enjamb* attribute for the **<l>** element, which alllows the presence or absence of enjambement to be signaled. The following lines demonstrate the use of the *enjamb* attribute to mark places where there is a discrepancy between the boundaries of the **<l>** elements and the syntactic structure of the verse (a discrepancy of some significance in some schools of verse):

```
<l enjamb=y>Un astrologue, un jour, se laissa choir</l>
<l>Au fond d'un puits.
```

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

```
<!-- 9.3:  Caesura                                          -->
<!-- The SEG element is defined in the linking tag set.     -->

<!ELEMENT caesura       - O  EMPTY                           >
<!ATTLIST caesura            %a.global;                      >
<!-- This fragment is used in sec. 9.1                      -->
```

9.4 Rhyme and Metrical Analysis

When the base tag set for verse is in use, the following additional attributes are available to record information about rhyme and metrical form:

met : contains a user-specified encoding for the conventional metrical structure of the element.
real : contains a user-specified encoding for the actual realization of the conventional metrical structure applicable to the element.

rhyme : specifies the rhyme scheme applicable to a group of verse lines.

These attributes may be attached to the **<lg>** element, to any of its numbered equivalents **<lg1>**, **<lg2>**, etc., to the higher-level text-division elements **<div>**, **<div1>**, etc., or to the **<body>** element itself if the whole of a text is in verse. In general, the attributes should be specified at the highest level possible; they may not however be specifiable at the highest level if some of the subdivisions of a text are in prose and others in verse. All these attributes may also be attached to the **<l>** and **<seg>** elements, but the default notation for the *rhyme* attribute has no defined meaning when specified on **<l>** or **<seg>**. The value for these attributes may take any form desired by the encoder; the nature of the notation used will determine how well the attribute values can be processed by automatic means.

The primary function of the metrical attributes is to encode the conventional metrical or rhyming structure within which the poet is working, rather than the actual prosodic realization of each line; the latter can be recorded using the *real* attribute, as further discussed below. There is no provision at this time, however, for encoding the particular realization of a rhyme pattern.

9.4.1 Sample Metrical Analyses

As a simple example of the use of these attributes, consider the following lines from Pope's "Essay on Criticism":

```
<text>
  <front> ... </front>
  <body met='-+|-+|-+|-+|-+/' rhyme='aa'>
    <lg1 n=1 type="verse paragraph">
      <l>'Tis hard to say, if greater Want of Skill</l>
      <l>Appear in <hi>Writing</> or in <hi>Judging</hi> ill;</l>
      <l>But, of the two, less dang'rous is th'Offence,</l>
      <l>To tire our <hi>Patience</hi>, than mis-lead
         our <hi>Sense</hi>:</l>
    <!-- ... -->
  </body>
</text>
```

The body of this text is written entirely in *heroic couplets*; each line is iambic pentameter (which, using a common notation, can be described with the formula "-+|-+|-+|-+|-+/", each "-" denoting a metrically unstressed syllable, each "+" a metrically stressed one, each "|" a foot boundary, and the "/" a line-end), and the couplets rhyme (which can be represented with the conventional formula "aa").

Because both rhyme pattern and metrical form are consistent throughout the poem, they may be most conveniently specified on the **<body>** element; the values given for the attributes will be inherited by any metrical unit contained within this **<body>** element, and must be interpreted in the appropriate way.

Since the notation used in the *met*, *real*, and *rhyme* attributes is user-defined, no binding description can be given of its details or of how its interpretation must proceed. (A default notation is provided for the *rhyme* attribute, which however the encoder can replace with another; see section 9.5, "Rhyme," on page 18.) It is expected, however, that software should be able to support these attributes in useful ways; the more intelligent the software is, and the more knowledge of metrics is built into it, the better it will be able to support these attributes. In the extract given above, for example, the *met* and *rhyme* attribute values specified on the **<body>** element are inherited directly by the **<lg>** elements nested within the body. Since the *met* value specifies the metrical form of a single verse line, the structure of the **<lg>** as a whole is understood to involve as many repetitions of the pattern as there are lines in the verse paragraph. The same attribute value, when inherited in turn by the **<l>** element, must be understood *not* to repeat. With sufficiently sophisticated software, segments within the line might even be understood as inheriting precisely that portion of the formula which applies to the segment in question; this will, however, be easier to accomplish for some languages than for others.

The *rhyme* attribute in this example uses the default notation to specify a rhyme scheme applicable only to pairs of lines. As elsewhere, the default notation for the *rhyme* attribute has no meaning for metrical units at the line level or below. In verse forms where line-internal rhyme is structurally significant, e.g. in some skaldic poetry, the default notation is incapable of expressing the required information, since the rhyme pattern may need to be specified for units smaller than the line. In such cases, a user-specified rhyme notation must be substituted for the default notation, or else the rhyme pattern must be described using some alternative method (e.g. by using the **<link>** mechanism described below).

The precise semantics of the *met* attribute and the inferences which software is expected or able to draw from it, are implementation-dependent; so are the semantics and processing of the *rhyme* attribute, when user-specified notations are used.

A formal definition of the significance of each component of the pattern given as the value of the *met* attribute may be provided in the **<metNotation>** element within the **<encodingDecl>** element in the TEI header (see section [5.3.6], “The Metrical Notation Description,” [in separate fascicle]). The encoder is free to invent any notation appropriate to his or her analytic needs, provided that it is adequately documented in this element. The notation may define metrical components using invented or traditional names (such as “iamb” or “hexameter”) or in terms of basic units such as codes for stressed or unstressed syllables, or a combination of the two.

The *real* (for “realization”) attribute may optionally be specified to indicate any deviation from the pattern defined by the *met* attribute which the encoder wishes to record. By default, the *real* attribute has the same value as the *met* attribute on the same element; it is only necessary to provide an explicit value when the realization differs in some way from the abstract metrical pattern. The tension between conventional metrical pattern and its realization may thus be recorded explicitly. For example, many readers of the above passage would stress the word “But” at the beginning of the third line rather than the word “of” following it, as the metrical pattern would normally require. This variation might be encoded as follows:

```
<l real='+-|-+|-+|-+|-+'>But of the two...
```

Where the *real* attribute is used to over-ride the default or conventional metrical pattern, it applies only to the element on which it is specified. The default pattern for any subsequent lines is unaffected.

As it happens, this particular kind of variation is very common in the English iambic pentameter;[3] an encoder might therefore choose to regard this not as an instance of a variant realization, but as an instance of a variant metrical form:

```
<l met='+-|-+|-+|-+|-+'>But of the two...
```

Alternatively, a different metrical notation might be defined, in which this kind of variation was permitted throughout the text.

In choosing whether to over-ride a metrical specification in this way or by using the *real* attribute, the encoder is required to determine whether the change is a systematic or conventional one (as in this example) or an occasional variation, perhaps for local effect. In the following example, from Goethe's "Auf dem See", the variation is a matter of local realization:

```
<lg1 type="chevy-chase-stanza"
     met="-+-+-+-+/-+-+-+"
     rhyme='ababcdcd'>
<l n=1>                  Und frische Nahrung, neues Blut
<l n=2 real="+--+-+">    Saug' ich aus freier Welt;
<l n=3 real="+--+-+-+">  Wie ist Natur so hold und gut,
<l n=4 real="---+-+">    Die mich am Busen haelt!
<l n=5>                  Die Welle wieget unsern Kahn
<l n=6>                  Im Rudertakt hinauf,
<l n=7>                  Und Berge, wolkig himmelan,
<l n=8>                  Begegnen unserm Lauf.
```

On the other hand, the famous inserted alexandrine in Pope's "Essay on Criticism", might be encoded as follows:

```
<l n=356> A needless alexandrine ends the song,
<l n=357 met='-+|-+|-+|-+|-+|-+' real='++|-+|-+|+-|++|-+'>
That, like a wounded snake, drags its slow length along.
</l>
```

Here the *met* attributes indicates that a different metrical convention (the alexandrine) is in force, while the *real* attribute indicates that there is a variation from that convention. As with many other aspects of metrical analysis, however, this is of necessity an entirely interpretive judgment.

9.4.2 Segment-Level versus Line-level Tagging

The examples given so far have encoded information about the realization of metrical conventions at the level of the whole verse-line. This has obvious advantages of simplicity, but the disadvantage that any deviation from metrical convention is not marked at its precise point of occurrence in the text. Greater precision may be achieved, but only at the cost of marking deviant metrical units explicitly. This may be done with the **<seg>** element, giving the variant realization as the value of the *real* attribute on that element. Using this method, the example given immediately above

---

[3] It even has a name: *anapaestic substitution*.

might be encoded as follows:

```
<l n=356> A need<seg1 type=foot n=2 real='--'>less
   a</seg1>lexandrine ends the song,
<l n=357 met='-+|-+|-+|-+|-+|-+'>
   <seg1 n=1 real='++'> That, like </seg1> a wounded snake,
   <seg1 n=4 real='++'> drags its </seg1>
   <seg1 n=5 real='++'> slow length </seg1>
   along.
</l>
```

The marking of the foot boundaries with the symbol "|" in the *met* attribute value of the **<l>** element allows the human reader, or a sufficiently intelligent software program, to isolate the correct portion of that attribute value as the default value for the same attribute on the **<seg1>** elements for feet, namely "-+". It is of course up to the encoder to decide whether or not to include the *n* attribute of **<seg>** here, and whether or not also to tag the feet in the line in which there is no deviation from the metrical convention. The ability of software to infer which foot is being marked, if not all are tagged, will depend heavily on the language of the text and the knowledge of prosody built into the software; the fuller and more explicit the markup, the easier it will be for software to handle it. It may prove useful, however, to mark metrical deviations in the manner shown, even if the available software is not sufficiently intelligent to scan lines without aid from the markup. Human readers who are interested in prosody may well be able to exploit the markup in useful ways even with less sophisticated software.

There are circumstances where it may also be useful to use the *met* attribute of **<seg>**. If we wish to identify the exact location of the different types of foot in the first line of Virgil's *Aeneid*, the text could be encoded as follows (for simplicity's sake the caesura has been omitted):

```
<l><seg1 type=foot met='+--'>Arma vi</>
   <seg1 met='+--'>rumque ca</>
   <seg1 met='++'>no Tro</>
   <seg1 met='++'>iae qui </>
   <seg1 met='+--'>primus ab</>
   <seg1 met='++'> oris</>
</l>
```

An appropriate value of the *met* attribute might also be encoded on the enclosing **<div>** or **<body>** element, to indicate that each foot may be made up of a dactyl or a spondee, so that the values given here for *met* at the level of the foot may be considered a series of local variations on this fundamental pattern; in cases like this, of course, the local variations may also be considered aspects of realization rather than of convention, in which case the *real* attribute may be used instead of *met*, if desired.

9.4.3 Metrical Analysis of Stanzaic Verse

The method described above may be used to encode quite complex verse forms, for instance various kinds of fixed-form stanzas. Let us take one of Dante's canzoni, in which each stanza except the last has the same combination of eleven-syllable and seven-syllable lines, and the same rhyme scheme:

```
<div0 type=canzone
      met='E/E/S/E/S/E/E/S/E/S/E/S/S/E/S/E/E/S/S/E/E'
      rhyme="abbcdaccbdceeffghhhgg">
<lg1 n=1 type=stanza>
<l n=1>Doglia mi reca nello core ardire
  <!-- ... -->
```

Here the *met* attribute specifies a metrical pattern for each of the twenty-one lines making up a stanza of the *canzone*. Each stanza inherits this definition from the parent **<div0>** element. The *rhyme* attribute specifies a rhyme scheme for each stanza, in the same way.

In the metrical notation used here, the letter “E” represents a line containing nine syllables which may or may not be metrically prominent, a tenth which is prominent and an optional non-prominent eleventh syllable. The letter “S” is used to represent a line containing five syllables which may or may not be metrically prominent, a sixth which is prominent and an optional non-prominent seventh syllable. A suitable definition for this notation might be given by a **<metNotation>** element like the following:

```
<metNotation type='met' pattern='((E|S)/)+)'>
  <symbol value='E' terminal=n>xxxxxxxxx+o
  <symbol value='S' terminal=n>xxxxx+o
  <symbol value='x'>metrically prominent or non-prominent
  <symbol value='+'>metrically prominent
  <symbol value='o'>optional non prominent
  <symbol value='/'>line division
</metNotation>
```

As noted above, the metrical pattern specified on the **<div0>** applies to each **<lg>** (stanza) element contained within the **<div0>**. In fact however, after seven stanzas of this type, there is a final stanza, known as a *commiato* or envoi, which follows a different metrical and rhyming scheme. The solution to this problem is simply to specify a new *met* attribute on the eighth stanza itself, which will override the default value inherited from parent **<div0>**, as follows:

```
<div0 met='.....'>
<!-- ... -->
<lg1>
  <!-- This line group inherits its met value -->
  <!-- from the containing <div0>             -->
  <l> ...
</lg1>
<lg1 type='commiato'
    met='E/S/S/E/S/E/E/S/S/E/E'
    rhyme="abbccdeeedd">
  <l n=1> Canzone, presso di qui
   <!-- ... -->
</lg1>
</div0>
```

Note that, in the same way as for the *real* attribute, over-riding of this kind does not affect subsequent elements at the same hierarchic level. Any **<lg1>** element following the *commiato* above would be assumed to use the same metrical and rhyming scheme as the one preceding the *commiato*. Moreover, although it is quite regular (in the sense that the last stanza of each *canzone* is a *commiato*), the over-riding must be specified for each case.

9.5 Rhyme

The *rhyme* attribute is used to specify the rhyme pattern of a verse form. Like the *met* attribute, it can be used with a user-specified notation documented by the **<metNotation>** element in the TEI header. Unlike *met*, however, the *rhyme* attribute has a default notation; if this default notation is used, no **<metNotation>** element need be given.

The default notation for rhyme offers the ability to record patterns of rhyming lines, using the traditional notation in which distinct letters stand for rhyming lines. For a work in rhyming couplets, like the Pope example above, the *rhyme* attribute simply specifies "aa", indicating that pairs of adjacent lines rhyme with each other. For a slightly more complex scheme, applicable to groups of four lines, in which lines 1 and 3 rhyme, as do lines 2 and 4, this attribute would have the value "abab". The traditional Spenserian stanza has the pattern "ababbcbcc", indicating that within each nine line stanza, lines 1 and 3 rhyme with each other, as do lines 2, 4, 5 and 7, and lines 6, 8 and 9.

Non-rhyming lines within such a group may be represented using a hyphen or an x, as in the following example:

```
<lg rhyme='AB-BBA'>
<l>The sunlight on the garden
<l>Hardens and grows cold,
<l>We cannot cage the minute
<l>Within its nets of gold
<l>When all is told
<l>We cannot beg for pardon.
</lg>
```

Note however that the default notation includes no specific way of recording "internal" rhyme, such as that between the end of the first line and the start of the second in this particular poem. For this, a special user-defined notation would need to be declared using the **<metNotation>** element in the header. Alternatively, rhyme, like alliteration or assonance, may be considered as a special form of "correspondence", and hence encoded using the mechanisms defined for that purpose in section 16.4, "Correspondence," [in separate fascicle].

To use the correspondence mechanisms to represent the complex rhyming pattern of the above example, we first need to delimit and identify each rhyming sequence within the text: the **<seg>** element may be used for this purpose, as follows:

```
<lg rhyme='AB-BBA'>
<l>The sunlight on the <seg id=A1>garden</seg>
<l><seg id=A2>Harden</seg>s and grows <seg id=B1>cold,</seg>
<l>We cannot cage the <seg id=C1>minute</seg>
<l>Wi<seg id=C2>thin it</seg>s nets of <seg id=B2>gold</seg>
<l>When all is <seg id=B3>told</seg>
```

```
<l>We cannot beg for <seg id=A3>pardon</seg>.
</lg>
```

Now that each rhyming word, or part-word, has been tagged and allocated an arbitrary identifier, the general purpose **<link>** element may be used to indicate which of the **<seg>** elements share the same rhyme, as follows:

```
<linkGrp type=rhyme>
<link targets='A1 A2 A3'>
<link targets='B1 B2 B3'>
<link targets='C1 C2'>
</linkGrp>
```

For further discussion of the **<link>** and **<linkGrp>** element, see section 16.4, “Correspondence,” [in separate fascicle].

9.6 Encoding Procedures For Other Verse Features

A number of procedures that may be of particular concern to encoders of verse texts are dealt with elsewhere in these guidelines. Some aspects of layout and physical appearance, especially important in the case of free verse, are dealt with in chapter 20, “Additional Tag Set for Transcription of Primary Sources,” [in separate fascicle]. Some initial recommendations for the encoding of phonetic or prosodic transcripts, which may be helpful in the analysis of sound structures in poetry, are to be found in chapter 11, “Base Tag Set for Transcriptions of Spoken Texts,” [in separate fascicle]; it may also be found convenient to use standard entity names (those proposed for the International Phonetic Alphabet suggest themselves) to mark positions of suprasegmentals such as primary and secondary stress, or other aspects of accentual structure.

As already indicated, chapter 16, “Segmentation and Alignment,” [in separate fascicle] contains much which will be found useful for the aligning of multiple levels of commentary and structure within verse analysis. Encoders of verse (as of other types of literary text) will frequently wish to attach identifying labels to portions of text that are not part of a system of hierarchical divisions, may overlap with one another, and/or may be discontinuous; for instance passages associated with particular characters, themes, images, allusions, topoi, styles, or modes of narration. Much of the computerized analysis of verse seems likely to require dividing texts up into blocks in this way. The **<span>** element discussed in 17.2, “Feature-Value Pairs,” [in separate fascicle] provides the means for doing this. Finally, the procedures for the tagging of feature structures, described in chapter 18, “Additional Tag Set for Feature Structures,” [in separate fascicle], provide a powerful means of encoding a wide variety of aspects of verse literature, including not only the metrical structures discussed above, but also such stylistic and rhetorical features as metaphor.

For other features it must for the time being be left to encoders to devise their own terminology. Elements such as **<metaphor tenor= ... vehicle= ...>** ... **</metaphor>** might well suggest themselves; but given the problems of definition involved, and the great richness of modern metaphor theory, it is clear that any such format, if pre-defined by these Guidelines, would have seemed objectionable to some and excessively restrictive to many. Leaving the choice of tagging terminology to individual encoders carries with it one vital corollary, however: the encoder must be utterly explicit, in the TEI header, about the methods of tagging used and the criteria and definitions on which they rest. Where no formal elements are currently proposed, such information may readily be given as simple prose description within the **<encodingDesc>** element defined in section 5.3, “The Encoding Description,” [in separate fascicle].

Part 7

Alphabetic Reference List of Tags

Class Name: **analysis**

Description: defines additional analytic attributes specifying alternate structural arrangements for the components of a text.

Attributes:

equiv (i.e. equivalent): specifies the identifiers of elements considered structurally equivalent to the one bearing the attribute.
Data type: IDREFS
Value: one or more valid SGML identifiers for elements in the current document separated by white space.
Default value: #IMPLIED
Example:

Remarks:
This attribute is optional.

exclude: specifies the identifiers of elements that are considered excluded from the structure when the element bearing this attribute is selected.
Data type: IDREFS
Value: one or more valid SGML identifiers for elements in the current document separated by white space.
Default value: #IMPLIED
Example:

Remarks:
This attribute is optional.

select: specifies which of several mutually exclusive elements actually occurs in a structure.
Data type: IDREFS
Value: one or more valid SGML identifiers for elements in the current document separated by white space.
Default value: #IMPLIED
Example:

Remarks:
This attribute is optional.

Part: additional tag set for analysis and interpretation.
Member of classes:
DTD file:
Discussed in 17, “Additional Tag Set for Simple Analysis,” [in separate fascicle].

<caesura>

Description: marks the point at which a metrical line may be divided.
Attributes: [None: global and inherited attributes only.]
Example:

```
<l>Hw&ae;t we Gar-Dena <caesura> in gear-dagum
<l>&t;eod-cyninga <caesura> &t;rym gefrunon,
<l>hy &d;a &ae;&t;elingas <caesura> ellen fremedon.
```

Remarks: .
Part: base tag set for verse texts.
Member of classes: vephr
DTD file:
Data description: Empty element
May occur within:
May contain: EMPTY
Declarations:

```
<!ELEMENT caesura       - O  EMPTY                          >
<!ATTLIST caesura            %a.global;                     >
```

Discussed in 9.3, “Components of the Verse Line,” on page 9.
This tag is optional.

Class Name: **certainty**

Description: defines additional attributes which are enabled for all elements when the additional tag set for certainty and uncertainty is selected.
Attributes: [None: global and inherited attributes only.]
Part: additional tag set for certainty.
Member of classes:
DTD file:
Discussed in 19, “Indications of Certainty and Responsibility,” [in separate fascicle].

Class Name: **chunk**

Description: includes all elements which can occur between, but not within, paragraphs and other chunks.
Attributes: [None: global and inherited attributes only.]
Remarks: This element class does not contain all those elements which can appear between chunks: the class *inter* contains a set of elements which can appear either within or between chunks. Unlike elements of that class, chunks cannot occur within chunks. In prose, this means the elements in this class can appear between but not within paragraphs.
Part: base tag set for common core features.
Member of classes: common
DTD file:
Discussed in 3.7, “Element Classes,” [in separate fascicle].

Class Name: **common**

Description: common chunk- and inter-level elements.

Attributes: [None: global and inherited attributes only.]

Remarks: This class defines the set of chunk- and inter-level elements available in all bases; it is used in defining the standard models *chunk.seq* and *specialPara* in the general and mixed bases.

Part: base tag set for common core features.

Member of classes:

DTD file:

Discussed in 3.7, “Element Classes,” [in separate fascicle].

Class Name: **comp.verse**

Description: component level elements unique to the base tag set for verse.

Attributes: [None: global and inherited attributes only.]

Remarks:

Part: base tag set for verse texts.

Member of classes:

DTD file: teivers2.ent

Discussed in 9.2, “Structural Divisions of Verse Texts,” on page 5.
Discussed in 3.7, “Element Classes,” [in separate fascicle].

Class Name: **declarable**

Description: elements which may be independently selected (using the special purpose *decls* attribute) from a candidate list of declarations within a TEI header.

Attributes:

> *default*: indicates whether or not this element is selected by default when its parent is selected.
>
> Data type: (YES|NO)
>
> > YES : This element is selected if its parent is selected
> >
> > NO : This element can only be selected explicitly, unless it is the only one of its kind, in which case it is selected if its parent is selected.
>
> Default value: NO
>
> This attribute is mandatory when applicable.

Remarks: The rules governing the association of declarable elements with individual parts of a TEI text are fully defined in chapter 25.3, “Associating Contextual Information with a Text,” [in separate fascicle]. Only one element of a particular type may carry the value *default=yes*.

Part: auxiliary tag set for TEI headers.

Member of classes:

DTD file:

Discussed in 25.3, “Associating Contextual Information with a Text,” [in separate fascicle].

Class Name: # divn

Description: structural elements which behave in the same way as divisions.

Attributes:

type: specifies a name conventionally used for this level of subdivision, e.g. “act”, “volume”, “book”, “section”, “canto”, etc.
Data type: CDATA
Value: any string of characters
Default value: #CURRENT
This attribute is recommended.

org: specifies how the content of the division is organized.
Data type: (composite | uniform)
Legal values are:
composite : composite content: i.e. no claim is made about the sequence in which the immediate contents of this division are to be processed, or their inter-relationships.
uniform : uniform content: i.e. the immediate contents of this element are regarded as forming a logical unit, to be processed in sequence.
Default value: uniform
This attribute is optional.

sample: indicates whether this division is a sample of the original source and if so, from which part.
Data type: (initial | medial | final | unknown | complete)
Legal values are:
initial : division lacks material present at end in source.
medial : division lacks material at start and end.
final : division lacks material at start.
unknown : position of sampled material within original unknown.
complete : division is not a sample.
Default value: complete
Remarks:
This attribute is optional.

part: specifies whether or not the division is fragmented by some other structural element, for example a speech crossing across two stanzas.
Data type: (Y | N | I | M | F)
Legal values are:
Y : the division is incomplete in some respect
N : either the division is complete, or no claim is made as to its completeness.
I : the initial part of an incomplete division
M : a medial part of an incomplete division
F : the final part of an incomplete division
Default value: N

Remarks: The values I, M, or F should be used only where it is clear how the division is to be reconstituted.
This attribute is mandatory when applicable.

Part: base tag set for common core features.
Member of classes:
DTD file:
Discussed in 7, “Default Text Structure,” [in separate fascicle].

Class Name: **enjamb** (i.e. enjambement)
Description: elements bearing the *enjamb* attribute.
Attributes:

enjamb (i.e. enjambement): indicates that the end of a verse line is marked by enjambement.
Data type: CDATA
Sample values include:
no : the line is end-stopped
yes : the line in question runs on into the next
weak : the line is weakly enjambed
strong : the line is strongly enjambed
Default value: #IMPLIED
Example:

Remarks: The usual practice will be to give the value “y” to this attribute when enjambement is being marked, or the values “weak” and “strong” if degrees of enjambement are of interest; if no value is given, however, the attribute does not default to a value of “no”; this allows the attribute to be omitted entirely when enjambement is not of particular interest.
This attribute is optional.

Part: base tag set for verse texts.
Member of classes:
DTD file: teivers2.ent
Discussed in 9.3, “Components of the Verse Line,” on page 9.

Class Name: **fs** (i.e. feature-structures)
Description: defines additional attributes for all elements, which are enabled when the additional tag set for feature-structure notation is selected.
Attributes: [None: global and inherited attributes only.]
Remarks: [No attributes declared yet.]
Part: additional tag set for analysis and interpretation.
Member of classes:

DTD file:
Discussed in 18, “Additional Tag Set for Feature Structures,” [in separate fascicle].

Class Name: **global**
Description: defines attributes common to all elements in the TEI encoding scheme.
Attributes:

id (i.e. identifier): provides a unique identifier for the element bearing the ID value.
Data type: ID
Value: any valid SGML name (in the reference concrete syntax, a name must begin with a letter and contain only letters, numeric digits, hyphen, and full stop).
Default value: #IMPLIED
Example:

```
<p id=names>Paragraph with the ID <mentioned>names</>....
<p id=dates>Paragraph with the ID <mentioned>dates</>....
```

Remarks: The *id* attribute may be used to specify a canonical reference for an element; see section 6.9, “Reference Systems,” [in separate fascicle].
This attribute is optional.

n (i.e. number, name, etc.): gives a number (or other label) for an element, which is not necessarily unique within the document.
Data type: CDATA
Value: any string of characters; often, but not necessarily, numeric.
Default value: #IMPLIED
Example:

Remarks: The *n* attribute may be used to specify the numbering of chapters, sections, list items, etc.; it may also be used in the specification of a standard reference system for the text.
This attribute is optional.

lang (i.e. language): indicates the language of the element content, usually using a two- or three-letter code from ISO 639.
Data type: IDREF
Value: The value must be the identifier specified for a writing system declaration declared in the TEI header, as described in section 5, “The TEI Header,” [in separate fascicle].
Default value: %INHERITED
Example:

```
<p lang=EN>The only surviving work by <person>Ari</>
(died 1148) is the ten-page <title
lang=IS>&Iacute;slendingab&oacute;k</> (<title
lang=LA>Libellus Islandorum</>), written in the early
twelfth century.
```

Remarks: If no value is specified for *lang*, the *lang* value for the immediately enclosing element is inherited; for this reason, a value should always be specified on the outermost element (**<tei.2>**).
This attribute is optional.

rend (i.e. rendition or presentation): indicates how the element in question was rendered or presented in the source text.
Data type: CDATA
Value: any string of characters; if the typographic rendition of a text is to be systematically recorded, a systematic set of values for the *rend* attribute should be defined.
Default value: #IMPLIED
Example:

Remarks: These Guidelines make no binding recommendations for the values of the *rend* attribute; the characteristics of visual presentation vary too much from text to text and the decision to record or ignore individual characteristics varies too much from project to project. Some potentially useful conventions are noted from time to time at appropriate points in the Guidelines.
This attribute is optional.

Remarks: The global attributes described here are made part of the attribute definition list declaration of each element by including the string “%a.global” in each such declaration. Some global attributes are made available when certain base or additional tag sets are selected; these are incorporated into the global attributes by references to the appropriate parameter entities. When the tag sets in question have not been selected, the parameter entities in question expand to the empty string.
Part: base tag set for common core features.
Member of classes: clterm cllink clana clfs clcert clspoken
DTD file:
Discussed in 3.5, “Global Attributes,” [in separate fascicle].

<l> (i.e. verse line)

Description: contains a single, possibly incomplete, line of verse.
Attributes:

part: specifies whether or not the line is metrically complete.
Data type: (Y | N | I | M | F)
Legal values are:
Y : the line is metrically incomplete

N : either the line is complete, or no claim is made as to its completeness
I : the initial part of an incomplete line
M : a medial part of an incomplete line
F : the final part of an incomplete line

Default value: N
Remarks: The values I, M, or F should be used only where it is clear how the line is to be reconstituted.
This attribute is mandatory when applicable.

Example:

```
<l part=Y met='-/-/-/-/-/'>
```

Part: base tag set for verse texts.
Member of classes: chunk metrical enjamb
DTD file:
Data description: contains character data or phrase level elements only
May occur within: lg lg1 lg2 lg3 lg4 lg5
May contain: %paraContent
Declarations:

```
<!ELEMENT l            - O  (%paraContent)                     >
<!ATTLIST l                 %a.global;
                            %a.enjamb;
                            %a.metrical;
          part              (Y | N | I | M | F) N              >
```

Discussed in 9.2, “Structural Divisions of Verse Texts,” on page 5.
This tag is mandatory when applicable.

<lg> (i.e. line group)

Description: contains a group of verse lines functioning as a formal unit e.g. a stanza, refrain, verse paragraph, etc.
Attributes: [None: global and inherited attributes only.]
Example:

```
<lg type=free>
<l>Let me be my own fool
<l>of my own making, the sum of it
</lg>
<lg>
<l>is equivocal.
<l>One says of the drunken farmer:
</lg>
<lg>
<l>leave him lay off it. And this is
<l>the explanation.
</lg>
```

Remarks:
Part: base tag set for common core features.
Member of classes: chunk metrical divn
DTD file:
Data description: contains verse lines or nested line groups only, possibly prefixed by a heading.
May occur within: lg
May contain: %m.divbot %m.divtop l lg
Declarations:

```
<!ELEMENT lg            - O  ((%m.divtop)*, (l | lg)+,
                              (%m.divbot)*)                    >
<!ATTLIST lg                 %a.global;
                             %a.divn;
                             %a.metrical;                      >
```

Discussed in 6.11.1, “Core Tags for Verse,” [in separate fascicle].
Discussed in 9.2, “Structural Divisions of Verse Texts,” on page 5.
This tag is optional.

<lg1> (i.e. line group)

Description: contains a first-level (i.e. largest) group of verse lines functioning as a formal unit e.g. a stanza, refrain, verse paragraph, etc.
Attributes: [None: global and inherited attributes only.]
Example:

```
<lg1 n='VIII' type=stanza>
   <lg2 type=sestet>
     <l>In the first year of Freedom's second dawn
     <l>Died George the Third; although no tyrant, one
     <l>Who shielded tyrants, till each sense withdrawn
     <l>Left him nor mental nor external sun:
     <l>A better farmer ne'er brushed dew from lawn,
     <l>A worse king never left a realm undone!
   </lg2>
   <lg2 type=couplet>
     <l>He died &dash; but left his subjects still behind,
     <l>One half as mad &dash; and t'other no less blind.
   </lg2>
</lg1>
```

Remarks:
Part: base tag set for verse texts.
Member of classes: chunk metrical compvers divn
DTD file:
Data description: contains verse lines or nested line groups only, possibly prefixed by a heading.
May occur within:
May contain: head l lg2
Declarations:

```
<!ELEMENT lg1           - o  (head?, (l | lg2)+)              >
<!ATTLIST lg1                %a.global;
                             %a.divn;
                             %a.metrical;                     >
```

Discussed in 9.2, “Structural Divisions of Verse Texts,” on page 5.
This tag is optional.

<lg2> (i.e. line group)

Description: contains a second-level (i.e. second largest) group of verse lines functioning as a formal unit e.g. a stanza, refrain, verse paragraph, etc.
Attributes: [None: global and inherited attributes only.]
Remarks:
Part: base tag set for verse texts.
Member of classes: chunk metrical divn
DTD file:
Data description: contains verse lines or nested line groups only, possibly prefixed by a heading.
May occur within: lg1
May contain: head l lg3
Declarations:

```
<!ELEMENT lg2           - o  (head?, (l | lg3)+)              >
<!ATTLIST lg2                %a.global;
                             %a.divn;
                             %a.metrical;                     >
```

Discussed in 9.2, “Structural Divisions of Verse Texts,” on page 5.
This tag is optional.

<lg3> (i.e. line group)

Description: contains a third-level (i.e. third largest) group of verse lines functioning as a formal unit e.g. a stanza, refrain, verse paragraph, etc.
Attributes: [None: global and inherited attributes only.]
Remarks:
Part: base tag set for verse texts.
Member of classes: chunk metrical divn
DTD file:
Data description: contains verse lines or nested line groups only, possibly prefixed by a heading.
May occur within: lg2
May contain: head l lg4
Declarations:

```
<!ELEMENT lg3           - o  (head?, (l | lg4)+)              >
<!ATTLIST lg3                %a.global;
```

```
                                      %a.divn;
                                      %a.metrical;                     >
```

Discussed in 9.2, “Structural Divisions of Verse Texts,” on page 5.
This tag is optional.

<lg4> (i.e. line group)

Description: contains a fourth-level (i.e. fourth largest) group of verse lines functioning as a formal unit e.g. a stanza, refrain, verse paragraph, etc.
Attributes: [None: global and inherited attributes only.]
Remarks:
Part: base tag set for verse texts.
Member of classes: chunk metrical divn
DTD file:
Data description: contains verse lines or nested line groups only, possibly prefixed by a heading.
May occur within: lg3
May contain: head l lg5
Declarations:

```
<!ELEMENT lg4           - o  (head?, (l | lg5)+)               >
<!ATTLIST lg4                %a.global;
                             %a.divn;
                             %a.metrical;                      >
```

Discussed in 9.2, “Structural Divisions of Verse Texts,” on page 5.
This tag is optional.

<lg5> (i.e. line group)

Description: contains a fifth-level (i.e. smallest) group of verse lines functioning as a formal unit e.g. a stanza, refrain, verse paragraph, etc.
Attributes: [None: global and inherited attributes only.]
Remarks:
Part: base tag set for verse texts.
Member of classes: chunk metrical divn
DTD file:
Data description: contains verse lines only, possibly prefixed by a heading.
May occur within: lg4
May contain: head l
Declarations:

```
<!ELEMENT lg5           - o  (head?, l+)                       >
<!ATTLIST lg5                %a.global;
                             %a.divn;
                             %a.metrical;                      >
```

Discussed in 9.2, “Structural Divisions of Verse Texts,” on page 5.
This tag is optional.

Class Name: **linking**

Description: defines additional attributes for hypertext and other linking, which are enabled for all elements when the additional tag set for linking is selected.

Attributes:

corresp (i.e. correspondents): points to elements that correspond to the current element in some way.
Data type: IDREFS
Value: one or more valid SGML identifiers for elements in the current document, separated by white space.
Default value: #IMPLIED
Example:

Remarks:
This attribute is optional.

synch (i.e. synchronous): points to elements that are synchronous with the current element.
Data type: IDREFS
Value: one or more valid SGML identifiers for elements in the current document, separated by white space.
Default value: #IMPLIED
Example:

Remarks:
This attribute is optional.

sameAs : points to an element that is the same as the current element.
Data type: IDREF
Value: a valid SGML identifier.
Default value: #IMPLIED
Example:

This attribute is optional.

copyOf : points to an element of which the current element is a copy.
Data type: IDREF
Value: a valid SGML identifier for an element in the current document.
Default value: #IMPLIED
Example:

Remarks: Any content of the current element should be ignored. Its true content is that of the element being pointed at.
This attribute is optional.

next : points to the next element of a virtual aggregate of which the current element is part.
Data type: IDREF
Value: a valid SGML identifier.
Default value: #IMPLIED
Example:

This attribute is optional.

prev : points to the previous element of a virtual aggregate of which the current element is part.
Data type: IDREF
Value: a valid SGML identifier.
Default value: #IMPLIED
Example:

This attribute is optional.

exclude : points to elements that are in exclusive alternation with the current element.
Data type: IDREFS
Value: a list of valid SGML identifiers.
Default value: #IMPLIED
Example:

This attribute is optional.

select : selects one or more alternants; if one alternant is selected, the ambiguity or uncertainty is marked as resolved. If more than one alternant is selected, the degree of ambiguity or uncertainty is marked as reduced by the number of alternants not selected.
Data type: IDREFS
Value: a list of valid SGML identifiers.
Default value: #IMPLIED
Example:

Remarks: This attribute should be placed on an element which is superordinate to all of the alternants from which the selection is being made.
This attribute is optional.

ana : points to elements which are analyses of the current element.
Data type: IDREFS
Value: a list of valid SGML identifiers.
Default value: #IMPLIED
Example:

Remarks: The current element should be a textual one. The element pointed at should be an analytic one.
This attribute is optional.

inst : points to instances of the current element.
Data type: IDREFS
Value: a list of valid SGML identifiers.
Default value: #IMPLIED
Example:

Remarks: The current element should be an analytic one. The element pointed at should be a textual one.
This attribute is optional.

Part: additional tag set for analysis and interpretation.
Member of classes:
DTD file:
Discussed in 16, “Segmentation and Alignment,” [in separate fascicle].

<metNotation>

Description: 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 metrically structured text (e.g. **<lg>**, **<l>**, or **<seg>**).
Attributes:

type: indicates whether the notation conveys the abstract metrical form, its actual prosodic realization, or the rhyme scheme, or some combination thereof.
Data type: NAMES
Value: One or more of the three attribute names *met*, *real*, or *rhyme*.
Default value: "MET REAL"
Example:

Remarks: By default, the **<metNotation>** element documents the notation used for metrical pattern and realization. It may also be used to document the notation used for rhyme scheme information; if not otherwise documented, the rhyme scheme notation defaults to the traditional “abab” notation.
This attribute is mandatory when applicable.

pattern: specifies a regular expression defining any value that is legal for this notation.
Data type: CDATA
Value: the value must be a valid expression for the PATTERN keyword as defined in the TEI extended pointer notation (see section 16.2.1.12, “TEI Extended Pointer Syntax,” [in separate fascicle]).
Default value: #IMPLIED

Example:

```
<metNotation pattern= '( (1|0)+  \|? /? )* ' >
<metNotation pattern= '((+-|-+)-+-+-+-+/)'
```

Remarks:
This attribute is optional.

Example:

```
<metNotation id=ip pattern= '((+-|-+)-+-+-+-+/)'
     <symbol value='+'>stressed syllable
     <symbol value='-'>unstressed syllable
     <symbol value='/'>metrical line boundary
</metNotation>
```

This example is intended for the far more restricted case typified by the Shakespearean iambic pentameter. Only metrical patterns containing exactly ten syllables, alternately stressed and unstressed, (except for the first two which may be in either order) to each metrical line can be expressed using this notation.

Remarks: The encoder may choose whether to define the notation formally or informally. However, the two methods may not be mixed.

Only usable within the header if the verse base is enabled.

Part: base tag set for TEI headers.
Member of classes: declabl
DTD file:
Data description: contains either a sequence of **<symbol>** elements or paragraphs. If one **<symbol>** is defined, then all the codes appearing within the *pattern* attribute should be documented, if the latter is specified.
May occur within:
May contain: %component.seq symbol
Declarations:

```
<!ELEMENT metNotation   - O  ((%component.seq) | (symbol+))     >
<!ATTLIST metNotation        %a.global;
                             %a.declarable;
          type               NAMES              "MET REAL"
          pattern            CDATA              #IMPLIED       >
```

Discussed in 9.4, “Rhyme and Metrical Analysis,” on page 12.
This tag is recommended.

Class Name: **metrical**
Description: elements carrying metrical information.
Attributes:

met: contains a user-specified encoding for the conventional metrical structure of the element.
Data type: CDATA
Value: May contain either a standard term for the kind of metrical unit (e.g. “hexameter”) or an encoded representation for the metrical pattern (e.g. “+--+-+-+-+-”). In either case, the notation used should be documented by a **<metNotation>** element within the **<encodingDesc>** of the associated header.
Default value: %INHERITED
Remarks: Where this attribute is not specified, the metrical pattern for the element concerned is understood to be inherited from its parent.
This attribute is recommended.

real: contains a user-specified encoding for the actual realization of the conventional metrical structure applicable to the element.
Data type: CDATA
Value: May contain either a standard term for the kind of metrical unit (e.g. “hexameter”) or an encoded representation for the metrical pattern (e.g. “+--+-+-+-+-”). In either case, the notation used should be documented by a **<metNotation>** element within the **<encodingDesc>** of the associated header.
Default value: #IMPLIED
Remarks: Where this attribute is not specified, the metrical realization for the element concerned is understood to be identical to that specified or implied for the *met* attribute.
This attribute is recommended when applicable.

rhyme: specifies the rhyme scheme applicable to a group of verse lines.
Data type: CDATA
Value: By default, the rhyme scheme is expressed as a string of alphabetic characters each corresponding with a rhyming line. Any non-rhyming lines should be represented by a hyphen or an X. Alternative notations may be defined as for *met* by use of the **<metNotation>** element in the TEI header.
Default value: #IMPLIED
Example:

```
<lg rhyme='ABABABCC'> ... </lg>
```

Remarks: When the default notation is used, it does not make sense to specify this attribute on any unit smaller than a line. Nor does the default notation provide any way to record internal rhyme, or to specify non-conventional rhyming practice. These extensions would require user-defined alternative notations.
This attribute is recommended.

Remarks:
Part: base tag set for verse texts.
Member of classes:
DTD file: teivers2.ent
Discussed in 9.4, “Rhyme and Metrical Analysis,” on page 12.

Class Name: **phrase**

Description: includes elements which can occur at the level of individual words or phrases.

Attributes: [None: global and inherited attributes only.]

Remarks: This class of elements can occur only within larger elements of the class *inter* or *chunk*. In prose, this means these elements can occur within paragraphs, list items, lines of verse, etc.

Part: base tag set for common core features.

Member of classes:

DTD file:

Discussed in 3.7, “Element Classes,” [in separate fascicle].

Class Name: **phrase.verse**

Description: phrase-level elements which may appear within verse only.

Attributes: [None: global and inherited attributes only.]

Remarks:

Part: additional tag set for verse texts.

Member of classes: phrase

DTD file: teivers2.ent

Discussed in 9.3, “Components of the Verse Line,” on page 9.

<seg> (i.e. arbitrary segment)

Description: contains any arbitrary phrase-level unit of text.

Attributes:

type: indicates what sort of segment is marked.
Data type: CDATA
Value: any string of characters.
Default value: #IMPLIED
Remarks: The *type* attribute may be used to provide any suitable classification for the **<seg>** elements tagged in a text. It should not be used to tag features for which other tags are defined in these Guidelines.
This attribute is optional.

part: specifies whether or not the segment is complete.
Data type: (Y | N | I | M | F)
Legal values are:
- Y : the segment is incomplete
- N : either the segment is complete, or no claim is made as to its completeness
- I : the initial part of an incomplete segment
- M : a medial part of an incomplete segment
- F : the final part of an incomplete segment

Default value: N

Remarks: The values I, M, or F should be used only where it is clear how the segment is to be reconstituted.
This attribute is mandatory when applicable.

Example:

```
<seg>When are you leaving?</seg>
<seg>Tomorrow.</seg>
```

Remarks: The **<seg>** element may be used at the encoder's discretion to mark any low-level segments of the text of interest for processing. One use of the element is to mark text features for which no appropriate markup is otherwise defined — i.e. as a simple extension mechanism. Another use is to provide an identifier for some segment which is to be pointed at by some other element — i.e. to provide a target, or a part of a target, for a **<ptr>** or other similar element.
Part: additional tag set for common core features.
Member of classes: clseg
DTD file:
Data description: May contain anything which may appear within a paragraph.
May occur within:
May contain: %paraContent
Declarations:

```
<!ELEMENT seg           - -  (%paraContent)                     >
<!ATTLIST seg                %a.global;
          type               CDATA                #IMPLIED
          part               (Y | N | I | M | F) N              >
```

Discussed in 6.12, “Arbitrary Segments,” [in separate fascicle].
Discussed in 11.4.1, “Segments,” [in separate fascicle].
This tag is optional.

Class Name: **seg**
Description: elements for arbitrary segmentation.
Attributes: [None: global and inherited attributes only.]
Remarks:
Part: base tag set for common core features.
Member of classes: phrase
DTD file:
Discussed in 6.12, “Arbitrary Segments,” [in separate fascicle].

Class Name: **spoken**
Description: defines additional attributes which are enabled for all elements when the base tag set for transcription of spoken texts is selected.
Attributes: [None: global and inherited attributes only.]

Part: additional tag set for spoken materials.
Member of classes:
DTD file:
Discussed in 11, “Base Tag Set for Transcriptions of Spoken Texts,” [in separate fascicle].

<symbol>

Description: 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:

value: specifies the character or character sequence being documented.
Data type: CDATA
Value: any available character or character sequence.
Default value: #REQUIRED
This attribute is required.

terminal: specifies whether the symbol is defined in terms of other symbols (*terminal*=*N*) or in prose (*terminal*=*Y*).
Data type: (Y|N)

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.

Default value: Y
This attribute is mandatory when applicable.

Example:

```
<symbol value='x'>a stressed syllable
<symbol value='o'>an unstressed syllable
<symbol value='A' terminal=N>xoo
```

Remarks:
Part: base tag set for TEI headers.
Member of classes:
DTD file:
Data description: May contain character data and phrase-level elements.
May occur within: metNotation
May contain: %phrase.seq;
Declarations:

```
<!ELEMENT symbol        - O  (%phrase.seq;)                    >
<!ATTLIST symbol             %a.global;
          value              CDATA               #REQUIRED
          terminal           (Y | N)             Y             >
```

Discussed in 9.4, “Rhyme and Metrical Analysis,” on page 12.

This tag is optional.

Entity Name: **tei.structure.dtd**
Description: defines the file in which the default text structure used by many base tag sets is defined.
Remarks:
Expansion: system 'teistr2.dtd'
Discussed in 3.2, “Core, Base, and Additional Tag Sets,” [in separate fascicle].

Class Name: **terminology** (i.e. global attributes for terminological data.)
Description: defines attributes for all elements in documents which use the base tag set for terminological data.
Attributes:

group: indicates the group (term and related elements) to which this element should be associated by specifying a string matching the *n* attribute value on an appropriate element.
Data type: CDATA
Value: any string matching the *n* attribute value on the **<term>** element to which the group is attached.
Default value: #IMPLIED
Example:

Remarks: The *group* attribute provides a specialized pointing mechanism for use within **<termEntry>** elements.
This attribute is optional.

grpPtr: indicates the group (term and related elements) to which this element should be associated by specifying its unique identifier, where this is available.
Data type: IDREF
Value: the value specified must match a value supplied as the value for an *id* attribute on some **<term>** element in the current SGML document.
Default value: #IMPLIED
Example:

Remarks: The *group* attribute provides a specialized pointing mechanism for use within **<termEntry>** elements.
This attribute is optional.

depend: indicates the parent element to which this element should be associated by specifying a string matching the *n* attribute value on an appropriate element.
Data type: CDATA
Value: any string matching the *n* attribute value on the element to which the dependent element is attached.
Default value: #IMPLIED
Example:

This attribute is optional.

depPtr: indicates the parent element to which this element should be associated by specifying its unique identifier, where this is available.
Data type: IDREF
Value: the value specified must match a value supplied as the value for an *id* attribute on some **<term>** element in the current SGML document.
Default value: #IMPLIED
Example:

Remarks:
This attribute is optional.

Remarks: The attributes shared by this element class are used for linking elements, possibly not adjacent in the record, which are related (e.g. a grammatical annotation and the term it describes). If no attribute is specified, the element is assumed to relate to the most recently specified **<term>** or **<otherForm>** element.
Part: base tag set for terminological data.
Member of classes:
DTD file:
Discussed in 14.2, “Tags for Terminological Data,” [in separate fascicle].
Discussed in 14.3.3, “Flat Term Entries Using Group and Depend Attributes,” [in separate fascicle].
Discussed in 14.4, “Overall Structure of Terminological Documents,” [in separate fascicle].

Usability Report and Comment

If you have comments of any kind on the current draft of the TEI Guidelines, please complete this form and return it to one of the following addresses:

C. M. Sperberg-McQueen, Computer Center (M/C 135), University of Illinois at Chicago, 1940 W. Taylor Room 124, Chicago IL 60612-7352, USA. Bitnet: U35395@UICVM, Internet: U35395@uicvm.uic.edu

Lou Burnard, Oxford University Computing Service, 13 Banbury Road, Oxford OX2 6NN, UK. Fax: +44 (865) 273275, Janet: LOU@UK.AC.OXFORD.VAX Internet: LOU@vax.oxford.ac.uk

Further copies of this form may be downloaded from LISTSERV @ UICVM (or listserv @ uicvm.uic.edu). Send a mail message containing the line 'GET TEIURC MEMO TEI-L'

Your name:
Your postal address:

Your e-mail address (if any):
Your occupation:
Your academic background:
Your immediate reactions to the Guidelines (please tick or cross one only):

Relevance to your present concerns or interests: high medium low
Importance to your future research plans: high medium low
Comprehensibility/usability of current draft: high medium low

Detailed technical comment: [Please feel free to comment on any part of the Guidelines or in general. Comments relating to specific parts of the text are most helpful if they include a section or chapter number]

Thank you!

Temporary Index