**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 29 TD
Text Encoding Initiative
Chicago, Oxford

© 1990, 1992 ACH, ACL, ALLC

September 9, 1993

Draft Version 2, September 9, 1993

# *Note*

This is the current draft (as of September 9, 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.

**Part I: Introduction**

1. About These Guidelines (released 13 March 1993)
2. Concise Summary of SGML (released 27 August 1993)
3. Structure of the TEI Document Type Definitions (released 7 April 1993)

**Part II: Core Tags and General Rules**

4. Characters and Character Sets (released 17 July 1992)
5. The TEI Header (released 19 August 1992)
6. Tags Available in All TEI Documents (released 10 Dec 1992)
7. Default Text Structure (released 26 March 1993)

**Part III: Base Tag Sets**

8. Base Tag Set for Prose (released 24 Oct 1992)
9. Base Tag Set for Verse (in preparation)
10. Base Tag Set for Drama (released 20 August 1993)
11. Base Tag Set for Transcriptions of Spoken Texts (released 23 Apr 1992)
12. Base Tag Set for Letters and Memoranda (in preparation)
13. Base Tag Set for Printed Dictionaries (in preparation)
14. Base Tag Set for Terminological Data (released 23 December 1993)
15. Composite Texts and Combining Bases (in preparation)

**Part IV: Additional Tag Sets**

16. Segmentation and Alignment (released 26 January 1993)
17. Additional Tag Set for Simple Analysis (in preparation)
18. Additional Tag Set for Feature Structures (in preparation)
19. CE: Indications of Certainty and Responsibility (in preparation)
20. Additional Tag Set for Transcription of Primary Sources (in preparation)
21. Additional Tag Set for Critical Apparatus (in preparation)
22. Additional Tags for Names and Dates (in preparation)
23. Networks, Graphs, Digraphs, and Trees (in preparation)
24. Formulae and Tables (TR4) (in preparation)
25. Additional Tag Set for Language Corpora (released 11 March 1993)

**Part V: Auxiliary Document Types**

26. SH The Independent Header (in preparation)
27. Writing System Declaration (in preparation)
28. Feature System Declaration (in preparation)
29. Tag Set Documentation (in preparation)

**Part VI: Technical Topics**

30. TEI Conformance (in preparation)
31. Modifying TEI DTDs (in preparation)
32. Rules for Negotiated and Non-Negotiated Interchange (in preparation)
33. Relationship of TEI to Other Standards (in preparation)
34. Markup for Non-Hierarchical Phenomena (in preparation)
35. Algorithm for Recognizing Canonical References (in preparation)

**Part VII: Alphabetical Reference List of Tags and Classes**

**Part VIII: Reference Material**

36. Full TEI Document Type Declarations (in preparation)
37. Standard Writing System Declarations (in preparation)
38. Sample Tag Set Documentation (in preparation)
39. Formal Grammar for the TEI-Interchange-Format Subset of SGML (released 4 Dec 1992)

## *Contents*

**Note** . . . . . **iii**

**Overall Table of Contents for TEI P2** . . . . . **v**

**Part 5: Auxiliary Document Types** . . . . . **1**

29 Tag Set Documentation . . . . . 3
29.1 The Tagdoc Documentation Element . . . . . 6
29.2 The AttList Documentation Element . . . . . 8
29.3 Element Classes . . . . . 10
29.4 Entity Documentation . . . . . 11
29.5 Example . . . . . 12

**Part 7: Alphabetic Reference List of Tags** . . . . . **13**

**Usability Report and Comment** . . . . . **51**

**Temporary Index** . . . . . **53**

# Part 5

## *Auxiliary Document Types*

# Chapter 29

## *Tag Set Documentation*

This chapter defines an auxiliary tag set which may be used for the documentation of new SGML elements, element classes, and entities. It is primarily intended for use by those wishing to extend or modify the content of these Guidelines in a conformant manner, as described in chapters 31, "Modifying TEI DTDs," [in separate fascicle] and 30, "TEI Conformance," [in separate fascicle]; it may however be of use for the documentation of any SGML or SGML-like encoding scheme. The elements described here are those used to document the TEI scheme itself, in part VII of the current document.[1]

Three distinct elements are used to document a tag set, the contents of each of which is described in more detail in the appropriate section of this chapter.

**<tsd>** : contains a set of elements documenting an SGML tag set.
**<tagDoc>** : documents the structure, content, and purpose of a single SGML element type. Attributes include:
*usage* : specifies the optionality of an attribute or element. Legal values are:
*req* : required
*mwa* : mandatory when applicable
*rec* : recommended
*rwa* : recommended when applicable
*opt* : optional
**<classDoc>** : contains reference information for one element class; either elements which appear together in SGML content models, or elements which share some common attribute.
**<entDoc>** : formally documents a single SGML entity within an encoding scheme.

In addition, the following phrase-level elements may be found useful for marking up occurrences of element names, etc., within the body of running text.

**<gi>** : contains the name (generic identifier) of an SGML element. Attributes include:
*TEI* : indicates whether this element is part of the TEI encoding scheme (i.e., defined in a TEI DTD fragment) or not. Legal values are:
*yes* : this element is part of the TEI scheme.
*no* : this element is not part of the TEI scheme.
**<att>** : contains the name of an attribute appearing within running text. Attributes include:
*TEI* : indicates whether this attribute is part of the TEI scheme (i.e., defined in a TEI DTD fragment) or not. Legal values are:
*yes* : this attribute is part of the TEI scheme.

---

[1] Mutatis mutandis: at the time this chapter is released, the reference materials being published are still using an earlier version of this tag set; like the rest of the Guidelines, they will be modified to use this published tag set as soon as the Guidelines are completed. -Eds.

*no* : this attribute is not part of the TEI scheme.

**<val>** : contains a single attribute value.

**<tag>** : contains text of a complete SGML start- or end-tag, possibly including attribute specifications, but excluding the opening and closing markup delimiter characters.

These four elements are included among the phrase-level elements available to any document using the auxiliary tag set defined in this chapter; to make them available to documents using other DTDs, they should be defined as extensions to the TEI tag sets, as described in chapter 31, “Modifying TEI DTDs,” [in separate fascicle].

As an example of the recommended use of these elements, we quote from a TEI working paper:

```
<p>The <gi>gi</gi> element is used to tag element names
when they appear in the text; the <gi>tag</gi> element
however is used to show how a tag as such might appear.  So
one might talk of an occurrence of the <gi>blort</gi>
element which had been tagged <tag>blort
type='runcible'</tag> ...  <tag>/blort</tag>.  The
<att>type</att> attribute may take any name token as value;
the default value is <val>spqr</val>, in memory of its
creator.
```

Note that end-tags marked with the **<tag>** element must include a leading slash (/) in the element content, as shown.

These elements and their components make up the auxiliary DTD for tag documentation, which is contained in the file *teitsd2.dtd*. This file has the following overall structure:

```
<!-- 29: File teitsd2.dtd: Auxiliary DTD for Tag Set          -->
<!-- Documentation                                            -->
<!-- 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.             -->


<!-- Embed entities for TEI generic identifiers.              -->

<!ENTITY % TEI.elementNames system 'teigis2.ent' >
%TEI.elementNames;
```

```
<!-- Embed entities for TEI keywords.                          -->

<!ENTITY % TEI.keywords.ent system 'teikey2.ent' >
%TEI.keywords.ent;

<!-- Define element classes for content models, shared        -->
<!-- attributes for element classes, and global attributes.   -->
<!-- (This all happens within the file teiclas2.ent.)         -->

<!ENTITY % TEI.elementClasses system 'teiclas2.ent' >
%TEI.elementClasses;

<!-- Embed the core tag set.                                   -->

<!ENTITY % TEI.core.dtd system 'teicore2.dtd' >
%TEI.core.dtd;

<!-- Define the top-level element for this DTD.               -->

<!ELEMENT tsd           - O  ((tagDoc | entDoc | classDoc)+)    >
<!ATTLIST tsd               %a.global;                           >

<!-- Define some additions for the phrase level tags.         -->

<!ELEMENT gi            - O  (#PCDATA)                         >
<!ATTLIST gi                %a.global;
          TEI               (yes | no)         yes             >
<!ELEMENT tag           - -  (#PCDATA)                         >
<!ATTLIST tag               %a.global;
          TEI               (yes | no)         yes             >
<!ELEMENT att           - -  (#PCDATA)                         >
<!ATTLIST att               %a.global;
          TEI               (yes | no)         yes             >
<!ELEMENT val           - O  (#PCDATA)                         >
<!ATTLIST val               %a.global;                         >

<!-- Finally we define the elements specific to this tag set. -->

<!-- ... declarations from section 29.1                       -->
<!--     (The tagDoc element)                                 -->
<!--     go here ...                                          -->
<!-- ... declarations from section 29.2                       -->
<!--     (Attribute documentation)                            -->
<!--     go here ...                                          -->
<!-- ... declarations from section 29.3                       -->
<!--     (Element classes)                                    -->
<!--     go here ...                                          -->
<!-- ... declarations from section 29.4                       -->
<!--     (Entity Documentation)                               -->
<!--     go here ...                                          -->
```

## 29.1 The Tagdoc Documentation Element

The **<tagDoc>** element is used to document an element type, together with its associated attributes. A completely specified **<tagDoc>** may comprise all of the following components in the order specified:

**<gi>** : contains the name (generic identifier) of an SGML element.
**<rs>** : contains a general purpose name or referencing string.
**<desc>** : contains a brief description of the purpose and application for an element, attribute, or attribute value.
**<attList>** : contains documentation for all the attributes associated with this element, as a series of **<attDef>** elements.
**<exemplum>** : contains a single example demonstrating the use of an element, together with optional paragraphs of commentary.
**<eg>** : contains a single example demonstrating the use of an element or attribute. If the example contains SGML markup, it must be enclosed within a marked CDATA section.
**<remarks>** : contains any commentary or discussion about the usage of an element, attribute, class, or entity not otherwise documented within the containing element.
**<part>** : specifies the module or part to which a particular element, element class, or entity belongs in a modular encoding scheme such as the TEI.
**<classes>** : specifies all the classes of which the documented element or class is a member or subclass.
**<files>** : specifies the name of the operating system file(s) within which this markup component is declared.
**<dataDesc>** : specifies the legal content of the element being documented, noting any semantic or application-dependent constraints, as well as constraints enforced by the SGML content model.
**<parents>** : lists elements which can directly contain this element.
**<children>** : lists the elements which this element may directly contain.
**<elemDecl>** : contains the full form of an SGML declaration for the element documented using the reference concrete syntax.
**<attlDecl>** : contains the SGML ATTLIST declaration associated with this element.
**<ptr>** : defines a pointer to another location in the current document in terms of one or more identifiable elements.
**<equiv>** : specifies an equivalent or comparable element in some other markup language.

If as is often the case the **<gi>** of an element is an abbreviation of a natural-language word or phrase, the **<rs>** element is used to provide the full form intended, which is important for mnemonic purposes. The **<ptr>** element is used to cross-refer to the relevant section(s) of any accompanying prose documentation.

As the content model for **<tagDoc>** makes clear, only the **<gi>**, **<desc>**, **<dataDesc>**, and **<elemDecl>** elements are mandatory components. For elements bearing attributes, the **<attList>** and **<attlDecl>** components are also required for TEI conformance. For consistency with the standard TEI reference materials, use of the **<classes>**, **<files>**, **<exemplum>**, **<parents>**, and **<children>** elements is strongly recommended. The only components of the **<tagDoc>** element which can appear more than once are the **<exemplum>**, **<ptr>**, and **<equiv>** elements. The order of components may not be changed.

Several examples of complete **<tagDoc>** elements are provided in section 38, "Sample Tag Set Documentation," [in separate fascicle].

The **<tagDoc>** element and its constituents are defined as follows:

```
<!-- 29.1:  The tagDoc element                                  -->
<!ELEMENT tagDoc        - -  (gi, rs?, desc, attList?,
                              exemplum*, remarks?, part?,
                              classes?, files?, dataDesc?,
                              parents?, children?, elemDecl,
                              attlDecl?, ptr*, equiv*)          >
<!ATTLIST tagDoc             %a.global;
          usage              (req | mwa | rec | rwa | opt)
                                                  opt           >
<!-- RS is defined in the core.                                 -->

<!-- GI was defined above.                                      -->

<!ELEMENT desc          - O  (%paraContent)                      >
<!ATTLIST desc               %a.global;                          >
<!ELEMENT attList       - O  (attDef*)                           >
<!ATTLIST attList            %a.global;                          >
<!ELEMENT exemplum      - -  (p*, eg, p*)                        >
<!ATTLIST exemplum           %a.global;                          >
<!ELEMENT eg            - -  (#PCDATA)                           >
<!ATTLIST eg                 %a.global;                          >
<!ELEMENT remarks       - O  (%component.seq)                    >
<!ATTLIST remarks            %a.global;                          >
<!ELEMENT part          - O  (#PCDATA)                           >
<!ATTLIST part               %a.global;                          >
<!ELEMENT classes       - O  EMPTY                               >
<!ATTLIST classes            %a.global;
          names              CDATA               #REQUIRED     >
<!ELEMENT files         - O  EMPTY                               >
<!ATTLIST files              %a.global;
          names              CDATA               #IMPLIED      >
<!ELEMENT dataDesc      - O  (%phrase.seq)                       >
<!ATTLIST dataDesc           %a.global;                          >
<!ELEMENT parents       - O  (#PCDATA)                           >
<!ATTLIST parents            %a.global;                          >
<!ELEMENT children      - O  (#PCDATA)                           >
<!ATTLIST children           %a.global;                          >
<!ELEMENT elemDecl      - O  (#PCDATA)                           >
<!ATTLIST elemDecl           %a.global;                          >
<!ELEMENT attlDecl      - -  (#PCDATA)                           >
<!ATTLIST attlDecl           %a.global;                          >
<!-- PTR is defined in the core.                                -->

<!ELEMENT equiv         - O  (%specialPara)                      >
<!ATTLIST equiv              %a.global;
          scheme             CDATA               #REQUIRED     >
<!-- This fragment is used in sec. 29                           -->
```

## 29.2 The AttList Documentation Element

The **<attList>** element is used to document information about a collection of attributes, either within a **<tagDoc>**, or within a **<classDoc>**. It consists of a series of **<attDef>** elements, each documenting a single attribute and each using an appropriate selection from the following elements:

**<attDef>** : contains the definition of a single attribute. Attributes include:
*usage* : specifies the optionality of an attribute or element. Legal values are:
*req* : required
*mwa* : mandatory when applicable
*rec* : recommended
*rwa* : recommended when applicable
*opt* : optional
**<attName>** : contains the name of the attribute being defined by an **<attDef>** element.
**<rs>** : contains a general purpose name or referencing string.
**<desc>** : contains a brief description of the purpose and application for an element, attribute, or attribute value.
**<dataType>** : specifies an SGML *declared value* for an attribute.
**<valList>** : contains a list of value and description pairs for an attribute. Attributes include:
*type* : specifies the extensibility of the list of attribute values specified. Legal values are:
*closed* : only the values specified are permitted.
*semi* : all the values specified should be supported, but other values are legal and software should have appropriate fallback processing for them.
*open* : the values specified are sample values only.
**<val>** : contains a single attribute value.
**<valDesc>** : specifies any semantic or syntactic constraint on the value that an attribute may take, additional to the information carried by the **<dataType>** element.
**<default>** : specifies the default declared value for an attribute.
**<eg>** : contains a single example demonstrating the use of an element or attribute. If the example contains SGML markup, it must be enclosed within a marked CDATA section.
**<remarks>** : contains any commentary or discussion about the usage of an element, attribute, class, or entity not otherwise documented within the containing element.
**<equiv>** : specifies an equivalent or comparable element in some other markup language.

If as is often the case the name of an attribute is an abbreviation of a natural-language word or phrase, the **<rs>** element is used to provide the full form intended.

It will be noted that several of these elements are used identically to document both elements and attributes. Specific to attributes are **<dataType>**, **<valList>**, **<valDesc>**, **<val>** and **<default>**. For any attribute documented in this way, either a **<valList>** or a **<valDesc>** must be supplied to specify the range of permitted values for an attribute. A **<valList>** should be used if the intended set of values can be enumerated; a **<valDesc>** if it cannot. A legal **<attDef>** specification must contain an **<attName>**, a **<desc>**, a **<dataType>**, either a **<valList>** or a **<valDesc>**, and a default; the other elements listed above are all optional.

The **<attList>** within a **<tagDoc>** is used to specify only the attributes which are specific to that particular element. Attributes which are shared by other elements, or by all elements, should be documented by an **<attList>** contained within a **<classDoc>** element, as described in section 29.3, "Element Classes," on page 10 below.

The following **<attList>** demonstrates some of the possibilities; for more detailed examples, refer to chapter 38, "Sample Tag Set Documentation," [in separate fascicle].

```
<attDef usage=opt>
<attName>type
<desc>describes the form of the list.
<dataType>CDATA
<valList type=semi>
  <val>ordered <desc>list items are numbered or lettered.</desc>
  <val>bulleted <desc>list items are marked with a
       <soCalled>bullet</soCalled> or other typographic
       device.</desc>
  <val>simple <desc>list items are not numbered or bulleted.</desc>
  <val>gloss  <desc>each list item glosses some term or
       concept, which is given by a <gi>label</gi> element
       preceding the list item.</desc>
</valList>
<default>simple
<remarks><p>The formal syntax of the element declarations allows
<gi>label</gi> tags to be omitted from lists tagged <tag>list
type=gloss</tag>; this is however a semantic error.
</attDef>
```

Those elements from the above list which are unique to attributes are declared as follows:

```
<!-- 29.2:  Attribute documentation                          -->
<!ELEMENT attDef        - O  (attName, rs?, desc, (dataType,
                             (valList | valDesc)?), default,
                             eg?, remarks?, equiv*)          >
<!ATTLIST attDef             %a.global;
          usage              (req | mwa | rec | rwa | opt)
                                                  opt        >
<!ELEMENT attName       - O  (#PCDATA)                        >
<!ATTLIST attName            %a.global;                       >
<!ELEMENT dataType      - O  (#PCDATA)                        >
<!ATTLIST dataType           %a.global;                       >
<!ELEMENT valList       - -  ((val, desc)*)                   >
<!ATTLIST valList            %a.global;
          type               (closed | semi | open)
                                                  open       >
<!ELEMENT valDesc       - O  (%phrase.seq)                    >
<!ATTLIST valDesc            %a.global;                       >
<!ELEMENT default       - O  (#PCDATA)                        >
<!ATTLIST default            %a.global;                       >
<!-- This fragment is used in sec. 29                       -->
```

## 29.3 Element Classes

The element **<classDoc>** is used to document an *element class*, as defined in section 3.7, "Element Classes," [in separate fascicle]. It has the following components:

**<classDoc>** : contains reference information for one element class; either elements which appear together in SGML content models, or elements which share some common attribute. Attributes include:
*type* : indicates whether this is an model class, an attribute class, or both. Sample values include:
*model* : members of this class appear in the same SGML content models
*atts* : members of this class share common attributes
*both* : members of this class share attributes and appear in the same SGML content models
**<class>** : specifies the name of an element class.
**<rs>** : contains a general purpose name or referencing string.
**<desc>** : contains a brief description of the purpose and application for an element, attribute, or attribute value.
**<attList>** : contains documentation for all the attributes associated with this element, as a series of **<attDef>** elements.
**<remarks>** : contains any commentary or discussion about the usage of an element, attribute, class, or entity not otherwise documented within the containing element.
**<part>** : specifies the module or part to which a particular element, element class, or entity belongs in a modular encoding scheme such as the TEI.
**<classes>** : specifies all the classes of which the documented element or class is a member or subclass.
**<ptr>** : defines a pointer to another location in the current document in terms of one or more identifiable elements.
**<equiv>** : specifies an equivalent or comparable element in some other markup language.

Of these elements, only the **<class>** and **<desc>** elements are required components. If present, the other elements must be given in the order specified, and only **<ptr>** and **<equiv>** may be repeated.

The attribute *type* is used to distinguish between "model" and "attribute" classes; for the former, a classDoc simply exists so that members of the class it documents may point to it (by specifying the value of its *id* attribute among the values specified by the *names* attribute of their **<classes>** component); for the latter, the classDoc additionally contains an **<attList>** which specifies the attributes shared by the members of the class. A class may perform both functions, of course.

Where a class inherits properties or attributes from some other class, the **<classes>** element may be used to indicate this fact. Membership of an attribute class can be inherited by any class, but model-only classes may not include attribute-only classes among their members. For further discussion of the TEI class system, see section 3.7, "Element Classes," [in separate fascicle].

For examples of attribute classes, please refer to chapter 38, "Sample Tag Set Documentation," [in separate fascicle].

The **<classDoc>** element and the elements unique to it are declared as follows:

```
<!-- 29.3:  Element classes                                    -->
<!ELEMENT classDoc      - O  (class, rs?, desc, attList?,
                              remarks?, part?, classes?, ptr*,
                              equiv*)                          >
<!ATTLIST classDoc           %a.global;
          type               (model | atts | both)
                                                 #IMPLIED      >
<!ELEMENT class         - O  (#PCDATA)                         >
<!ATTLIST class              %a.global;                        >
<!-- all other constituents are defined above                 -->

<!-- This fragment is used in sec. 29                         -->
```

## 29.4 Entity Documentation

The **<entDoc>** element is used to document any entity not otherwise documented by the elements described in this chapter. Its chief uses are to provide systematic documentation for parameter entities used within TEI DTD fragments (for example, those used to enable different components of the TEI DTD, or to describe common content models), but it may be used for any purpose. It has the following components:

**<entDoc>** : formally documents a single SGML entity within an encoding scheme. Attributes include:
  *type* : indicates whether this is a general or a parameter entity. Legal values are:
    *pe* : parameter entity
    *ge* : general entity

**<entName>** : contains the full name of an entity, excluding the percent sign in the case of a parameter entity.

**<rs>** : contains a general purpose name or referencing string.

**<desc>** : contains a brief description of the purpose and application for an element, attribute, or attribute value.

**<remarks>** : contains any commentary or discussion about the usage of an element, attribute, class, or entity not otherwise documented within the containing element.

**<string>** : contains the intended expansion for the entity documented by an **<entdoc>** element, enclosed by quotation marks.

**<ptr>** : defines a pointer to another location in the current document in terms of one or more identifiable elements.

**<equiv>** : specifies an equivalent or comparable element in some other markup language.

As in the other documentation elements, **<rs>** is used in cases when the entity name itself is an abbreviation, to give the full natural-language word or phrase abbreviated, and **<ptr>** is used to give cross-references to any prose documentation. Of the elements named, only **<entName>**, **<desc>** and **<string>** are required components. If present, the other elements must be given in the order specified, and only **<xref>** and **<equiv>** may be repeated.

For examples, please refer to chapters 38, “Sample Tag Set Documentation,” [in separate fascicle] and 3, “Structure of the TEI Document Type Definitions,” [in separate fascicle].

The **<entDoc>** element and the elements unique to it are declared as follows:

```
<!-- 29.4:  Entity Documentation                              -->
<!ELEMENT entDoc        - -  (entName, rs?, desc, remarks?,
                              string, ptr*, equiv*)             >
<!ATTLIST entDoc             %a.global;
          type               (pe | ge)          #REQUIRED      >
<!ELEMENT entName       - O  (#PCDATA)                          >
<!ATTLIST entName            %a.global;                         >
<!ELEMENT string        - -  (#PCDATA)                          >
<!ATTLIST string             %a.global;                         >
<!-- All other constituents are defined above                 -->

<!-- This fragment is used in sec. 29                         -->
```

## 29.5 Example

As described above, a tag set documentation file contains a series of **<tagDoc>**, **<classDoc>**, and **<entDoc>** elements, enclosed within a **<tsd>** element. The DTD for this auxiliary document type is contained in file *teitsd2.dtd*.

A complete tag set documentation file documenting the imaginary elements **<blort>** and **<granfalloon>** might thus have the following appearance:

```
<!DOCTYPE tsd system 'teitsd2.dtd'>
<tsd>
<tagDoc><gi>blort</gi>
 <desc>Contains a ...</desc>
 <elemDecl>- O (%phrase.seq)</elemDecl>
</tagDoc>
<tagDoc><gi>granfalloon</gi>
 <desc>Contains an unsupported assertion or fantastic
 accusation.</>
 <elemDecl>- - (%paraContent)</>
</tagDoc>
</tsd>
```

## 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: I wish I understood what this was for.
This attribute is optional.

Part: additional tag set foranalysis and interpretation.
Member of classes:
DTD file:

Discussed in 17, “Additional Tag Set for Simple Analysis,” [in separate fascicle].

# <att>

Description: contains the name of an attribute appearing within running text.

Attributes:

*TEI*: indicates whether this attribute is part of the TEI scheme (i.e., defined in a TEI DTD fragment) or not.
Data type: (yes|no)
Legal values are:
yes : this attribute is part of the TEI scheme.
no : this attribute is not part of the TEI scheme.

Default value: yes

This attribute is optional.

Example:

```
The TEI defines three <soCalled>global</soCalled> attributes;
their names are <att>id</att>, <att>rend</att> and <att>n</att>.
<att TEI=NO>type</att> is not among them.
```

Remarks:
Part: auxiliary tag set fortag set declarations.
Member of classes: sgmlkw
DTD file:
Data description: May contain a valid SGML name only.
May occur within:
May contain: #PCDATA
Element Declaration:

```
<!ELEMENT att           - -  (#PCDATA)                        >
```

Attribute List Declaration:

```
<!ATTLIST att               %a.global;
          TEI               (yes | no)          yes           >
```

Discussed in 29, “Tag Set Documentation,” on page 3.
This tag is optional.

# <attDef> (i.e. attribute definition)

Description: contains the definition of a single attribute.

Attributes:

*usage*: specifies the optionality of an attribute or element.
Data type: (req|mwa|rec|rwa|opt)
Legal values are:
req : required
mwa : mandatory when applicable
rec : recommended
rwa : recommended when applicable
opt : optional

Default value: opt

This attribute is optional.

Example:

```
 <attDef usage=rec>
<attName>type</attName>
<desc>specifies a name conventionally used
for this level of subdivision, e.g. <q>act</q>,
<q>volume</q>, <q>book</q>, <q>section</q>, <q>canto</q>, etc.</desc>
<dataType>CDATA
<valDesc>any string of characters</valDesc>
<default>#CURRENT
</attDef>
```

Part: auxiliary tag set fortag set declarations.
Member of classes:
DTD file:
Data description:
May occur within: attList
May contain: attName dataType default desc eg equiv remarks rs valDesc valList
Element Declaration:

```
<!ELEMENT attDef        - O  (attName, rs?, desc, (dataType,
                             (valList | valDesc)?), default,
                             eg?, remarks?, equiv*)            >
```

Attribute List Declaration:

```
<!ATTLIST attDef            %a.global;
          usage             (req | mwa | rec | rwa | opt)
                                                  opt          >
```

Discussed in 29.2, “The AttList Documentation Element,” on page 8.
This tag is mandatory when applicable.

# <attlDecl> (i.e. attList declaration)

Description: contains the SGML ATTLIST declaration associated with this element.
Attributes: [None: global and inherited attributes only.]
Example: If the opening delimiter is parsed by the SGML parser, an error will result; it should therefore be represented by an entity reference, thus:

```
<attlDecl>
 &lt;!ATTLIST blort
              id       ID       #REQUIRED
              rend     CDATA    #IMPLIED   >
</attlDecl>
```

Alternatively, the entire element content may be enclosed in a CDATA marked section:

```
<attlDecl>
 <![ CDATA [
 <!ATTLIST blort
              id       ID       #REQUIRED
              rend     CDATA    #IMPLIED   >
 ];]>
</attlDecl>
```

Part: auxiliary tag set fortag set declarations.
Member of classes:
DTD file:
Data description: contains the full text of an SGML ATTLIST declaration; to avoid parsing errors, the opening delimiter must be given as an entity reference, or else the entire content of the element should be enclosed in a CDATA marked section.
May occur within: tagDoc
May contain: #PCDATA
Element Declaration:

```
<!ELEMENT attlDecl      - -  (#PCDATA)                          >
```

Attribute List Declaration:

```
<!ATTLIST attlDecl           %a.global;                         >
```

Discussed in 29.1, “The Tagdoc Documentation Element,” on page 6.
This tag is required.

# <attList>

Description: contains documentation for all the attributes associated with this element, as a series of **<attDef>** elements.
Attributes: [None: global and inherited attributes only.]
Example:

```
<attList>
   <attDef> ... </attDef>
   <attDef> ... </attDef>
</attList>
```

Remarks:
Part: auxiliary tag set fortag set declarations.
Member of classes:
DTD file:
Data description: contains a series of **<attDef>** elements only
May occur within: classDoc tagDoc
May contain: attDef
Element Declaration:

```
<!ELEMENT attList       - O  (attDef*)                          >
```

Attribute List Declaration:

```
<!ATTLIST attList            %a.global;                         >
```

Discussed in 29.1, “The Tagdoc Documentation Element,” on page 6.
This tag is mandatory when applicable.

# <attName> (i.e. attribute name)

Description: contains the name of the attribute being defined by an **<attDef>** element.
Attributes: [None: global and inherited attributes only.]
Example:

```
<attName>type</attName>
```

Remarks:
Part: auxiliary tag set fortag set declarations.
Member of classes:
DTD file:
Data description: must be a valid SGML name.
May occur within: attDef
May contain: #PCDATA
Element Declaration:

```
<!ELEMENT attName       - O  (#PCDATA)                          >
```

Attribute List Declaration:

```
<!ATTLIST attName           %a.global;                          >
```

Discussed in 29.2, “The AttList Documentation Element,” on page 8.
This tag is mandatory when applicable.

# 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.
Member of classes:
DTD file:
Discussed in 19, “CE: Indications of Certainty and Responsibility,” [in separate fascicle].

# <children>

Description: lists the elements which this element may directly contain.
Attributes: [None: global and inherited attributes only.]
Example:

```
<children>emph, hi, name, address, #PCDATA
```

Part: auxiliary tag set fortag set declarations.
Member of classes:
DTD file:
Data description: the list may contain a description or a list of names separated by commas or spaces.
May occur within: tagDoc
May contain: #PCDATA
Element Declaration:

```
<!ELEMENT children      - O  (#PCDATA)                          >
```

Attribute List Declaration:

```
<!ATTLIST children          %a.global;                          >
```

Discussed in 29.1, “The Tagdoc Documentation Element,” on page 6.
This tag is required.

# <class>

Description: specifies the name of an element class.
Attributes: [None: global and inherited attributes only.]
Example:

```
<classDoc>
 <class>phrase</class>
 <!-- ... -->
</classDoc>
```

Part: auxiliary tag set fortag set declarations.
Member of classes:
DTD file:
Data description: In TEI compatible encoding systems, the class name should be chosen so that a valid SGML name may be made from it. For attribute classes, this name is made by prefixing the class name with the string “a.”; for model classes, the string “m.” is used.
May occur within: classDoc
May contain: #PCDATA
Element Declaration:

```
<!ELEMENT class         - O  (#PCDATA)                          >
```

Attribute List Declaration:

```
<!ATTLIST class              %a.global;                         >
```

Discussed in 29.3, “Element Classes,” on page 10.
This tag is mandatory when applicable.

# <classDoc>

Description: contains reference information for one element class elements which appear together in SGML content models, or elements either which share some common attribute.

Attributes:

*type*: indicates whether this is an model class, an attribute class, or both.
Data type: (model | atts | both)

model : members of this class appear in the same SGML content models
atts : members of this class share common attributes
both : members of this class share attributes and appear in the same SGML content models

Default value: #IMPLIED

This attribute is required.

Example:

```
<classDoc type=model ID=seg>
<class>seg</class>
<desc>elements for arbitrary segmentation.</desc>
<ptr target=COse>
</classDoc>
```

Part: auxiliary tag set fortag set declarations.
Member of classes:
DTD file:
Data description:
May occur within: tsd
May contain: attList class classes desc equiv part ptr remarks rs
Element Declaration:

```
<!ELEMENT classDoc      - O  (class, rs?, desc, attList?,
                              remarks?, part?, classes?, ptr*,
                              equiv*)                          >
```

Attribute List Declaration:

```
<!ATTLIST classDoc           %a.global;
          type               (model | atts | both)
                                                 #IMPLIED      >
```

Discussed in 29.3, “Element Classes,” on page 10.
This tag is recommended when applicable.

# <classes>

Description: specifies all the classes of which the documented element or class is a member or subclass.

Attributes:

*names*: lists the identifiers of all classes of which the documented element or class is a member or subclass, possibly using parentheses to indicate inheritance.
Data type: CDATA
Value: a list of class names separated by spaces or commas, and optionally enclosed by parentheses; each name should be the class name specified for some element class in the scheme being documented or modified.
Default value: #REQUIRED

This attribute is optional.

Example:

```
<classes names='hqinter (phrase)'>
```

This **<classes>** element indicates that the element documented is a member of the class "hqinter", which is itself a subclass of the class "phrase".

```
<classes names='hqinter'>
```

This **<classes>** element indicates that the element documented is a member of the class "hqinter", but gives no indication that "hqinter" is itself a subclass of "phrase".

Part: auxiliary tag set fortag set declarations.
Member of classes:
DTD file:
Data description: Empty
May occur within: classDoc tagDoc
May contain: EMPTY
Element Declaration:

```
<!ELEMENT classes       - O  EMPTY                              >
```

Attribute List Declaration:

```
<!ATTLIST classes           %a.global;
          names             CDATA               #REQUIRED      >
```

Discussed in 29, "Tag Set Documentation," on page 3.
This tag is recommended when applicable.

## 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 forcommon core features.
Member of classes:
DTD file:
Discussed in 3.7, "Element Classes," [in separate fascicle].

## Class Name: data

Description: phrase-level elements containing names, dates, numbers, measures, and similar data.
Attributes: [None: global and inherited attributes only.]
Remarks:
Part: base tag set forcommon core features.
Member of classes: phrase
DTD file:
Discussed in 6.4, “Names, Numbers, Dates, Abbreviations, and Addresses,” [in separate fascicle].

# <dataDesc>

Description: specifies the legal content of the element being documented, noting any semantic or application-dependent constraints, as well as constraints enforced by the SGML content model.
Attributes: [None: global and inherited attributes only.]
Example:

```
<dataDesc>A declarative sentence, an adjectival phrase, or the word
EMPTY.
```

Part: auxiliary tag set fortag set declarations.
Member of classes:
DTD file:
Data description: A declarative sentence, an adjectival phrase, or the word EMPTY.
May occur within: tagDoc
May contain: %phrase.seq
Element Declaration:

```
<!ELEMENT dataDesc      - O  (%phrase.seq)                    >
```

Attribute List Declaration:

```
<!ATTLIST dataDesc           %a.global;                       >
```

Discussed in 29, “Tag Set Documentation,” on page 3.
This tag is optional.

# <dataType>

Description: specifies an SGML *declared value* for an attribute.
Attributes: [None: global and inherited attributes only.]
Example:

```
<dataType>CDATA
```

Part: auxiliary tag set fortag set declarations.
Member of classes:
DTD file:
Data description: must be a legal SGML keyword, notation name or a name token group.
May occur within: attDef
May contain: #PCDATA
Element Declaration:

```
<!ELEMENT dataType      - O  (#PCDATA)                          >
```

Attribute List Declaration:

```
<!ATTLIST dataType           %a.global;                         >
```

Discussed in 29.2, “The AttList Documentation Element,” on page 8.
This tag is required.

# <default>

Description: specifies the default declared value for an attribute.
Attributes: [None: global and inherited attributes only.]
Example:

```
<default>#IMPLIED
```

Part: auxiliary tag set fortag set declarations.
Member of classes:
DTD file:
Data description: any legal SGML declared value or TEI-defined keyword
May occur within: attDef
May contain: #PCDATA
Element Declaration:

```
<!ELEMENT default       - O  (#PCDATA)                          >
```

Attribute List Declaration:

```
<!ATTLIST default            %a.global;                         >
```

Discussed in 29.2, “The AttList Documentation Element,” on page 8.
This tag is required.

# <desc> (i.e. description)

Description: contains a brief description of the purpose and application for an element, attribute, or attribute value.
Attributes: [None: global and inherited attributes only.]
Example:

```
<desc>contains a brief description of the purpose and application for
an element, attribute, attribute value, class, or entity.
```

Part: auxiliary tag set fortag set declarations.
Member of classes:
DTD file:
Data description: TEI convention requires that this be expressed as a finite clause, begining with an active verb.
May occur within: attDef classDoc entDoc tagDoc valList
May contain: %paraContent
Element Declaration:

```
<!ELEMENT desc          - O  (%paraContent)                    >
```

Attribute List Declaration:

```
<!ATTLIST desc              %a.global;                         >
```

Discussed in 29.1, “The Tagdoc Documentation Element,” on page 6.
This tag is required.

# <eg>

Description: contains a single example demonstrating the use of an element or attribute. If the example contains SGML markup, it must be enclosed within a marked CDATA section.
Attributes: [None: global and inherited attributes only.]
Example:

This example shows how an **<eg>** element may contain a CDATA marked section.

```
<exemplum>
<p>The <gi>term</gi> element may be used to mark any
technical term:</p>
<eg><![ CDATA [
This <term>recursion</> is giving me a headache.
];]>
</eg>
</exemplum>
```

Part: auxiliary tag set fortag set declarations.
Member of classes:

DTD file:
Data description:
May occur within: attDef exemplum
May contain: #PCDATA
Element Declaration:

```
<!ELEMENT eg            - -  (#PCDATA)                          >
```

Attribute List Declaration:

```
<!ATTLIST eg                %a.global;                          >
```

Discussed in 29, “Tag Set Documentation,” on page 3.
This tag is mandatory when applicable.

# <elemDecl> (i.e. element declaration)

Description: contains the full form of an SGML declaration for the element documented using the reference concrete syntax.
Attributes: [None: global and inherited attributes only.]
Example:

The contents of this element are usually surrounded by a CDATA marked section, which prevents the markup-declaration-open delimiter (“”) at the beginning of the element content from being recognized by the SGML parser (and thus causing an error).

```
<elemDecl>
<![ CDATA [
<!ELEMENT list - - (head?, ((item*)
                  | (head.label?, head.item?,
                  (label, item)+)))  >
];]>
</elemDecl>
```

The markup-declaration-open delimiter can however also be given as (or be interrupted by) an entity reference:

```
<elemDecl>
&lt;!ELEMENT list - - (head?, ((item*)
                  | (head.label?, head.item?,
                  (label, item)+)))  >
</elemDecl>
```

Part: auxiliary tag set fortag set declarations.
Member of classes:
DTD file:

Data description: The declaration must be enclosed with a CDATA marked section, or else the opening delimiter must be given by an entity reference.
May occur within: tagDoc
May contain: #PCDATA
Element Declaration:

```
<!ELEMENT elemDecl      - O  (#PCDATA)                          >
```

Attribute List Declaration:

```
<!ATTLIST elemDecl           %a.global;                         >
```

Discussed in 29, “Tag Set Documentation,” on page 3.
This tag is optional.

# <entDoc> (i.e. entity documentation)

Description: formally documents a single SGML entity within an encoding scheme.

Attributes:

*type*: indicates whether this is a general or a parameter entity.
Data type: (pe | ge)
Legal values are:
pe : parameter entity
ge : general entity

Default value: #REQUIRED

This attribute is required.

Example:

```
<entDoc id=extPtr type=pe>
<entName>extPtr</entName>
<rs>extended-pointer expression</rs>
<desc>used as the declared value of an attribute, indicates that all
values of that attribute must be valid expressions in the TEI extended
pointer notation.
<string>'CDATA'</string>
<ptr target=SAxr1>
</entDoc>
```

Part: auxiliary tag set fortag set declarations.
Member of classes:
DTD file:
Data description:
May occur within: tsd
May contain: desc entName equiv ptr remarks rs string

Element Declaration:

```
<!ELEMENT entDoc        - -  (entName, rs?, desc, remarks?,
                              string, ptr*, equiv*)             >
```

Attribute List Declaration:

```
<!ATTLIST entDoc             %a.global;
          type               (pe | ge)          #REQUIRED      >
```

Discussed in 29.4, “Entity Documentation,” on page 11.
This tag is recommended when applicable.

# <entName> (i.e. entity name)

Description: contains the full name of an entity, excluding the percent sign in the case of a parameter entity.
Attributes: [None: global and inherited attributes only.]
Example:

```
<entName>component.seq
```

Part: auxiliary tag set fortag set declarations.
Member of classes:
DTD file:
Data description: Must be a valid SGML name.
May occur within: entDoc
May contain: #PCDATA
Element Declaration:

```
<!ELEMENT entName       - O  (#PCDATA)                          >
```

Attribute List Declaration:

```
<!ATTLIST entName            %a.global;                         >
```

Discussed in 29.4, “Entity Documentation,” on page 11.
This tag is mandatory when applicable.

# <equiv>

Description: specifies an equivalent or comparable element in some other markup language.

Attributes:

*scheme*: names the markup language or encoding scheme
Data type: CDATA
Value: any phrase identifying a markup language
Default value: #REQUIRED

This attribute is required.

Example:

```
<tagDoc>
   <gi>p</gi>
   <!-- ... -->
   <equiv mlang=LaTeX>\para</>
</tagDoc>
```

Remarks:
Part: auxiliary tag set fortag set declarations.
Member of classes:
DTD file:
Data description: If the other markup language uses SGML-significant characters, a CDATA marked section should be used.
May occur within: attDef classDoc entDoc tagDoc
May contain: %specialPara
Element Declaration:

```
<!ELEMENT equiv         - O  (%specialPara)                    >
```

Attribute List Declaration:

```
<!ATTLIST equiv              %a.global;
          scheme             CDATA               #REQUIRED      >
```

Discussed in 29.1, “The Tagdoc Documentation Element,” on page 6.
This tag is optional.

# <exemplum>

Description: contains a single example demonstrating the use of an element, together with optional paragraphs of commentary.
Attributes: [None: global and inherited attributes only.]
Example:

```
<exemplum>
 <p>The <gi>name</gi> element can be used for both
 personal names and place names:</p>
 <eg>
   <![ CDATA [
   <q>My dear <name type=person>Mr. Bennet</name>,</q>
   said his lady to him one day, <q>have you heard that
```

```
  <name type=place>Netherfield Park</name> is let
  at last?</q>
  ];]>
 </eg>
 <p>As shown, the <att>type</att> attribute may be used
 to distinguish the one from the other.
</exemplum>
```

Note that an explicit end-tag must be supplied for the paragraph immediately preceding the **<eg>** element within an **<exemplum>**, to prevent the **<eg>** from being mistaken for part of the paragraph.

Part: auxiliary tag set fortag set declarations.
Member of classes:
DTD file:
Data description:
May occur within: tagDoc
May contain: eg p
Element Declaration:

```
<!ELEMENT exemplum      - -  (p*, eg, p*)                    >
```

Attribute List Declaration:

```
<!ATTLIST exemplum          %a.global;                       >
```

Discussed in 29, “Tag Set Documentation,” on page 3.
This tag is recommended when applicable.

# <files>

Description: specifies the name of the operating system file(s) within which this markup component is declared.

Attributes:

*names*: supplies the names of one or more files.
Data type: CDATA
Value: a file identifier
Default value: #IMPLIED

This attribute is required.

Example:

```
<files names=TEIdict2>
```

Remarks:
Part: auxiliary tag set fortag set declarations.
Member of classes:

DTD file:
Data description: Empty element.
May occur within: tagDoc
May contain: EMPTY
Element Declaration:

```
<!ELEMENT files         - O  EMPTY                              >
```

Attribute List Declaration:

```
<!ATTLIST files             %a.global;
          names             CDATA               #IMPLIED       >
```

Discussed in 29.1, "The Tagdoc Documentation Element," on page 6.
This tag is optional.

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 foranalysis and interpretation.
Member of classes:
DTD file:
Discussed in 18, "Additional Tag Set for Feature Structures," [in separate fascicle].

<gi> (i.e. generic identifier)
Description: contains the name (generic identifier) of an SGML element.

Attributes:

*TEI*: indicates whether this element is part of the TEI encoding scheme (i.e., defined in a TEI DTD fragment) or not.
Data type: (yes|no)
Legal values are:
yes : this element is part of the TEI scheme.
no : this element is not part of the TEI scheme.
Default value: yes

This attribute is optional.

Example:

```
A  <gi tei=n>blort</gi> stalked the prairies.
The element <gi>sourceDesc</gi> is missing.
```

Part: auxiliary tag set fortag set declarations.
Member of classes: sgmlkw
DTD file:
Data description: TEI naming rules require lower case letters for all identifiers, except for names derived from multi-word phrases, in which case all but the first word is capitalized.
May occur within: tagDoc
May contain: #PCDATA
Element Declaration:

```
<!ELEMENT gi            - O  (#PCDATA)                          >
```

Attribute List Declaration:

```
<!ATTLIST gi                %a.global;
          TEI               (yes | no)          yes             >
```

Discussed in 29, “Tag Set Documentation,” on page 3.
This tag is mandatory when applicable.

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 "REFID= not defined" target=HD>.
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 forcommon core features.
Member of classes: clterm cllink clana clfs clcert clspoken
DTD file:
Discussed in 3.5, “Global Attributes,” [in separate fascicle].

## 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): specifies the identifiers of corresponding elements.
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.

*link* (i.e. link): specifies the identifiers of elements that may be linked with this one to form an aggregate.
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.

*partof*: specifies the identifier of a **<join>** element in the current document representing an aggregate of which this element is a component part.
Data type: IDREF
Value: a valid SGML identifier attached to a **<join>** element.
Default value: #IMPLIED
Example:

Remarks:
This attribute is optional.

*prev* (i.e. previous): specifies the identifier of an element which may be linked with the current element to form an aggregate, and which immediately precedes it in the aggregate.
Data type: IDREF
Value: a valid SGML identifier for an element in the current document.
Default value: #IMPLIED
Example:

Remarks:
This attribute is optional.

*next* (i.e. next): specifies the identifier of an element which may be linked with the current element to form an aggregate, and which immediately follows it in the aggregate.
Data type: IDREF
Value: a valid SGML identifier for an element in the current document.
Default value: #IMPLIED
Example:

Remarks:
This attribute is optional.

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

## Class Name: loc

Description: elements used for purposes of location and reference.
Attributes: [None: global and inherited attributes only.]
Remarks:
Part: base tag set forcommon core features.
Member of classes: phrase
DTD file:
Discussed in 6.6, “Simple Links and Cross References,” [in separate fascicle].

## Class Name: names

Description: elements with proper nouns as content.
Attributes:

*key*: provides an alternative identifier for the object being named, such as a database record key.
Data type: CDATA
Value: any string
Default value: #IMPLIED
Example:

```
<name type='place'>Montaillou</name>
is not a large parish.
At the time of the events which led to <name type='person' key=BXII>
Fournier's</name> investigations,
the local population consisted of between 200 and
250 inhabitants.
```

Remarks: The value may be a unique identifier from a database, or simply a more explicit name for the referent. Its purpose is only to record an identification; if the analysis leading to the identification is to be recorded as well, the analytic tags described in chapter "REFID= not defined" target=FS> should be used in addition or instead.
This attribute is optional.

*reg* (i.e. regularization): gives a normalized or regularized form of the proper name
Data type: CDATA
Value: Any string of characters.
Default value: #IMPLIED
Example:

```
At the time of the events which led to
<name type=person reg='Benedict XII, Pope of Avignon
(Jacques Fournier)'>Fournier's</name> investigations,
the local population consisted of between 200 and
250 inhabitants.
```

Remarks: In providing a "regularized" form, no claim is made that the form in the source text is incorrect; the regularized form is simply that chosen as the main form for purposes of unifying variant forms under a single heading.
This attribute is optional.

Remarks:
Part: base tag set forcommon core features.
Member of classes:
DTD file:
Discussed in 6.4.1, "Referring Strings," [in separate fascicle].

# <parents>

Description: lists elements which can directly contain this element.
Attributes: [None: global and inherited attributes only.]

Example:

```
<parents>head, title
```

Part: auxiliary tag set fortag set declarations.
Member of classes:
DTD file:
Data description: a list of valid SGML names separated by commas or spaces.
May occur within: tagDoc
May contain: #PCDATA
Element Declaration:

```
<!ELEMENT parents       - O  (#PCDATA)                          >
```

Attribute List Declaration:

```
<!ATTLIST parents            %a.global;                         >
```

Discussed in 29.1, “The Tagdoc Documentation Element,” on page 6.
This tag is required.

# <part>

Description: specifies the module or part to which a particular element, element class, or entity belongs in a modular encoding scheme such as the TEI.
Attributes: [None: global and inherited attributes only.]
Example:

```
<part>Base tag set for print dictionaries</part>
```

Remarks:
Part: auxiliary tag set fortag set declarations.
Member of classes:
DTD file:
Data description: A descriptive phrase identifying the module concerned.
May occur within: classDoc tagDoc
May contain: #PCDATA
Element Declaration:

```
<!ELEMENT part          - O  (#PCDATA)                          >
```

Attribute List Declaration:

```
<!ATTLIST part               %a.global;                         >
```

Discussed in 29.1, “The Tagdoc Documentation Element,” on page 6.
This tag is optional.

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 forcommon core features.
Member of classes:
DTD file:
Discussed in 3.7, “Element Classes,” [in separate fascicle].

# <ptr>

Description: defines a pointer to another location in the current document in terms of one or more identifiable elements.

Attributes:

*target*: specifies the destination of the pointer as one or more SGML identifiers
Data type: IDREFS
Value: Each value specified must be the same as that specified as value for an *ID* attribute for some other element in the current SGML document.
Default value: #REQUIRED

This attribute is required.

Example:

```
<ptr target="P2 P5" resp=LB type=s>
<ptr target='D1' type='defn'>
```

Remarks:
Part: base tag set forcommon core features.
Member of classes: clloc pointer
DTD file:
May occur within: classDoc entDoc tagDoc
May contain: EMPTY
Element Declaration:

```
<!ELEMENT ptr           - O  EMPTY                              >
```

Attribute List Declaration:

```
<!ATTLIST ptr               %a.global;
                            %a.pointer;
          target            IDREFS              #REQUIRED      >
```

Discussed in 6.6, “Simple Links and Cross References,” [in separate fascicle].
This tag is optional.

Class Name: pointer
Description: elements which point to other elements, using IDREFs.

Attributes:

*type*: categorizes the pointer in some respect, using any convenient set of categories.
Data type: CDATA
Value: The type should indicate the intended function of the pointer, or the rhetorical relationship between its source and target.
Default value: #IMPLIED

This attribute is optional.

*resp* : specifies the creator of the pointer.
Data type: CDATA
Value: any string of characters, usually the initials or name of the creator.
Default value: #IMPLIED
This attribute is optional.

*crdate* : specifies when the pointer was created.
Data type: CDATA
Value: any string representing a date.
Default value: #IMPLIED
This attribute is optional.

*targType* : specifies the kinds of elements to which this pointer may point.

Data type: NAMES
Value: A list of names of SGML elements defined in the DTD of the current document.
Default value: #IMPLIED
Remarks: If this attribute is supplied, every element specified as a target must be of one or the other of the types specified. An application may choose whether or not to report failures to satisfy this constraint as errors, but may not access an element of the right identifier but the wrong type.

This attribute is optional.

*targOrder* : where more than one identifier is supplied as the value of the *target* attribute, this attribute specifies whether the order in which they are supplied is significant.
Data type: (Y | N | U)
Legal values are:
Y : Yes: the order in which IDREFs are specified as the value of a *target* attribute should be followed when combining the targeted elements.
N : No: the order in which IDREFs are specified as the value of a *target* attribute has no significance when combining the targeted elements.
U : Unspecified: the order in which IDREFs are specified as the value of a *target* attribute may or may not be significant.

Default value: U

This attribute is optional.

*evaluate* : specifies the intended meaning when the target of a pointer is itself a pointer.
Data type: (all | one | none)

all : if the element pointed to is itself a pointer, then the target of that pointer will be taken, and so on, until an element is found which is not a pointer.
one : if the element pointed to is itself a pointer, then its target (whether a pointer or not) is taken as the target of this pointer.
none : no further evaluation of targets is carried out beyond that needed to find the element specified in the pointer's target.

Default value: #IMPLIED

Remarks: If no value is given, the application program is responsible for deciding (possibly on the basis of user input) how far to trace a chain of pointers.

This attribute is optional.

Remarks:
Part: base tag set forcommon core features.
Member of classes:
DTD file:
Discussed in 6.6, “Simple Links and Cross References,” [in separate fascicle].

# <remarks>

Description: contains any commentary or discussion about the usage of an element, attribute, class, or entity not otherwise documented within the containing element.
Attributes: [None: global and inherited attributes only.]
Example:

```
<remarks><p>This element is probably redundant.
```

Remarks: As defined in ODD, must contain paragraphs; should be special.para
Part: auxiliary tag set fortag set declarations.
Member of classes:
DTD file:
Data description: Contains at least one paragraph, unless it is empty.
May occur within: attDef classDoc entDoc tagDoc
May contain: %component.seq
Element Declaration:

```
<!ELEMENT remarks       - O  (%component.seq)                  >
```

Attribute List Declaration:

```
<!ATTLIST remarks            %a.global;                        >
```

Discussed in 29.1, “The Tagdoc Documentation Element,” on page 6.
This tag is optional.

## <rs> (i.e. referencing string)

Description: contains a general purpose name or referencing string.
Attributes:

*type*: indicates more specifically the object referred to by the referencing string. Values might include “person”, “place”, “ship”, “element” etc.
Data type: CDATA

Default value: #IMPLIED
This attribute is mandatory when applicable.

Example:

```
<q>My dear <rs type=person>Mr. Bennet</rs>, </q>
said <rs type=person>his lady</rs> to him one day,
<q>have you heard that <rs type=place>Netherfield
Park</rs> is let at last?</q>
```

Remarks:
Part: additional tag set forcorrespondence.
Member of classes: data names
DTD file:

May occur within: attDef classDoc entDoc tagDoc
May contain: %phrase.seq
Element Declaration:

```
<!ELEMENT rs            - -  (%phrase.seq)                    >
```

Attribute List Declaration:

```
<!ATTLIST rs                 %a.global;
                             %a.names;
          type               CDATA               #IMPLIED       >
```

Discussed in 22.1, “Personal Names,” [in separate fascicle].
Discussed in 6.4.1, “Referring Strings,” [in separate fascicle].
This tag is optional.

## Class Name: sgmlKeywords (i.e. SGML keywords)

Description: elements whose content is an SGML identifier or tag of some sort (generic identifier of an element type, name of an attribute, etc.).
Attributes: [None: global and inherited attributes only.]
Remarks: These elements are defined in the auxiliary tag set for tag set documentation; they may be useful in writing SGML documentation as well.
Part: auxiliary tag set fortag set declarations.
Member of classes: phrase
DTD file:
Discussed in 29, “Tag Set Documentation,” on page 3.

## 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 forspoken materials.
Member of classes:
DTD file:
Discussed in 11, “Base Tag Set for Transcriptions of Spoken Texts,” [in separate fascicle].

# <string>

Description: contains the intended expansion for the entity documented by an **<entdoc>** element, enclosed by quotation marks.
Attributes: [None: global and inherited attributes only.]
Example:

```
<string>"the choice of quotes may'nt be insignificant"</string>
```

System entities should include the *system* keyword within the content of this element, as shown:

```
<string>SYSTEM 'teiclas2.ent'</>
```

Remarks: The content of this element is the SGML *entity text* for the entity, including any keywords.
Part: auxiliary tag set fortag set declarations.
Member of classes:
DTD file:
Data description: any sequence of character data
May occur within: entDoc
May contain: #PCDATA
Element Declaration:

```
<!ELEMENT string        - -  (#PCDATA)                          >
```

Attribute List Declaration:

```
<!ATTLIST string            %a.global;                          >
```

Discussed in 29.4, “Entity Documentation,” on page 11.
This tag is required.

# <tag>

Description: contains text of a complete SGML start- or end-tag, possibly including attribute specifications, but excluding the opening and closing markup delimiter characters.

Attributes:

*TEI*: indicates whether this tag is valid within the TEI scheme or not.
Data type: (yes|no)

yes : this is a valid TEI tag.
no : this is not a valid TEI tag.

Default value: yes

This attribute is optional.

Example:

```
Mark the start of each italicised phrase with
a <tag>hi rend=it</tag> tag, and its end
with a <tag>/hi</tag> tag.
```

Part: auxiliary tag set fortag set declarations.
Member of classes: sgmlkw
DTD file:
Data description: contains only characters legal between mark up delimiter characters.
May occur within:
May contain: #PCDATA
Element Declaration:

```
<!ELEMENT tag           - -  (#PCDATA)                          >
```

Attribute List Declaration:

```
<!ATTLIST tag               %a.global;
          TEI               (yes | no)          yes             >
```

Discussed in 29, “Tag Set Documentation,” on page 3.
This tag is optional.

# <tagDoc>

Description: documents the structure, content, and purpose of a single SGML element type.

Attributes:

*usage*: specifies the optionality of an attribute or element.
Data type: (req|mwa|rec|rwa|opt)
Legal values are:
req : required
mwa : mandatory when applicable
rec : recommended
rwa : recommended when applicable
opt : optional
Default value: opt
This attribute is optional.

Example:

Part: auxiliary tag set fortag set declarations.
Member of classes:
DTD file:
Data description:

May occur within: tsd
May contain: attlDecl attList children classes dataDesc desc elemDecl equiv exemplum files gi parents part ptr remarks rs
Element Declaration:

```
<!ELEMENT tagDoc        - -  (gi, rs?, desc, attList?,
                              exemplum*, remarks?, part?,
                              classes?, files?, dataDesc?,
                              parents?, children?, elemDecl,
                              attlDecl?, ptr*, equiv*)          >
```

Attribute List Declaration:

```
<!ATTLIST tagDoc            %a.global;
          usage             (req | mwa | rec | rwa | opt)
                                                  opt           >
```

Discussed in 29.1, “The Tagdoc Documentation Element,” on page 6.
This tag is optional.

## Entity Name: TEI.core.dtd (i.e. TEI core)

Description: identifies the file containing element and attribute list declarations for the TEI core elements.
Remarks: This entity is included in all TEI DTDs.
Expansion: system 'teicore2.dtd'
Discussed in 3.2, “Core, Base, and Additional Tag Sets,” [in separate fascicle].

## Entity Name: TEI.elementClasses

Description: file containing the parameter entity declarations which define element classes for content models, attributes shared among elements of a class, and global attributes.
Remarks:
Expansion: system 'teiclas2.ent'
Discussed in 3.6, “The TEI2.DTD File,” [in separate fascicle].
Discussed in 3.7, “Element Classes,” [in separate fascicle].

## Entity Name: TEI.elementNames

Description: file containing parameter entity declarations for all generic identifiers of the encoding scheme.

Remarks: The parameter entities in this file all take the same form as the two shown below:

```
<!ENTITY % n.div0 'div0' >
<!ENTITY % n.div1 'div1' >
```

Element and attribute-list declarations in the DTDs refer to the parameter entity *n.div1*, not directly to the generic identifier *div1*. As a result, the declarations will function as desired even if a new generic identifier is substituted (e.g. “caput” for “div1” and “liber” for “div0”:

```
<!ENTITY % n.div0 'liber' >
<!ENTITY % n.div1 'caput' >
```

This allows generic identifiers to be renamed conveniently, to provide names in languages other than English, or to provide shorter names than those documented here. See further chapter 31, “Modifying TEI DTDs,” [in separate fascicle].

Expansion: system 'teigis2.ent'
Discussed in 3.8.2, “Parameter Entities for Element Generic Identifiers,” [in separate fascicle].

Entity Name: TEI.keywords.ent
Description: file containing the parameter entity declarations for TEI keywords.
Remarks: The keywords defined in this file define non-SGML data types (e.g. ISO dates) and control the selection of base and additional tag sets.
Expansion: system 'teikey2.ent'
Discussed in 3.8.3, “Parameter Entities for TEI Keywords,” [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 forterminological 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].

## <tsd> (i.e. tag set documentation)

Description: contains a set of elements documenting an SGML tag set.
Attributes: [None: global and inherited attributes only.]
Example:

```
<tsd>
<tagDoc> ... </tagDoc>
<entDoc type=pe> ... </entDoc>
<tagDoc> ... </tagDoc>
</tsd>
```

Remarks: This element is required.
Part: auxiliary tag set fortag set declarations.
Member of classes:
DTD file:
Data description: Contains any mixture of **<tagDoc>**, **<entDoc>**, and **<classDoc>** elements.
May occur within:
May contain: classDoc entDoc tagDoc
Element Declaration:

```
<!ELEMENT tsd           - O  ((tagDoc | entDoc | classDoc)+)    >
```

Attribute List Declaration:

```
<!ATTLIST tsd               %a.global;                          >
```

Discussed in 29, “Tag Set Documentation,” on page 3.
This tag is required.

# <val> (i.e. value)

Description: contains a single attribute value.
Attributes: [None: global and inherited attributes only.]
Example:

```
<val>unknown</val>
```

Remarks:
Part: auxiliary tag set fortag set declarations.
Member of classes: sgmlkw
DTD file:
Data description:
May occur within: valList
May contain: #PCDATA
Element Declaration:

```
<!ELEMENT val           - O  (#PCDATA)                          >
```

Attribute List Declaration:

```
<!ATTLIST val               %a.global;                          >
```

Discussed in 29.2, “The AttList Documentation Element,” on page 8.
This tag is optional.

# <valDesc> (i.e. value description)

Description: specifies any semantic or syntactic constraint on the value that an attribute may take, additional to the information carried by the **<dataType>** element.
Attributes: [None: global and inherited attributes only.]
Example:

```
<valDesc>must point to another <gi>align</gi>
   element logically preceding this
   one.</valDesc>
```

Part: auxiliary tag set fortag set declarations.
Member of classes:
DTD file:
Data description: contains any sequence of phrase level elements or pcdata
May occur within: attDef
May contain: %phrase.seq
Element Declaration:

```
<!ELEMENT valDesc       - O  (%phrase.seq)                    >
```

Attribute List Declaration:

```
<!ATTLIST valDesc            %a.global;                       >
```

Discussed in 29.2, “The AttList Documentation Element,” on page 8.
This tag is recommended when applicable.

# <valList> (i.e. value list)

Description: contains a list of value and description pairs for an attribute.

Attributes:

*type*: specifies the extensibility of the list of attribute values specified.
Data type: (closed | semi | open)

Legal values are:
closed : only the values specified are permitted.
semi : all the values specified should be supported, but other values are legal and software should have appropriate fallback processing for them.
open : the values specified are sample values only.

Default value: open

This attribute is optional.

Example:

```
<valList>
 <val>req<desc>required
 <val>mwa<desc>mandatory when applicable
 <val>rec<desc>recommended
 <val>rwa<desc>recommended when applicable
 <val>opt<desc>optional
</valList>
```

Part: auxiliary tag set fortag set declarations.
Member of classes:
DTD file:
Data description:
May occur within: attDef
May contain: desc val
Element Declaration:

```
<!ELEMENT valList       - -  ((val, desc)*)                    >
```

Attribute List Declaration:

```
<!ATTLIST valList           %a.global;
          type              (closed | semi | open)
                                                 open          >
```

Discussed in 29.2, “The AttList Documentation Element,” on page 8.
This tag is mandatory when applicable.

# *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*

att (tag) ... 3, 14
attDef (tag) ... 6, 8, 10, 14, 16-17
attlDecl (tag) ... 6, 16
attList (tag) ... 6, 8-10, 16
attName (tag) ... 8, 17
atts ... 10

both ... 10

children (tag) ... 6, 18
chunk ... 37
chunk.seq ... 21
class (tag) ... 10, 18
classDoc (tag) ... 3, 8-12, 19, 47
classes (tag) ... 6, 10, 20-21
closed ... 8

dataDesc (tag) ... 6, 22
dataType (tag) ... 8, 22, 48
declared value ... 8, 22
default (tag) ... 8, 23
desc (tag) ... 6, 8, 10-11, 24
div1 ... 45

eg (tag) ... 6, 8, 24, 29
elemDecl (tag) ... 6, 25
element ... 10
entDoc (tag) ... 3, 11-12, 26, 41, 47
entity text ... 42
entName (tag) ... 11, 27
equiv (tag) ... 6, 8, 10-11, 27
exemplum (tag) ... 6, 28-29

files (tag) ... 6, 29

ge ... 11
gi (tag) ... 3, 6, 30
group (attribute) ... 46

id (attribute) ... 10, 31, 37, 46
inter ... 37

join (tag) ... 33

lang (attribute) ... 32

model ... 10
mwa ... 3, 8

n (attribute) ... 32, 45-46
n.div1 ... 45
names (attribute) ... 10
no ... 3-4

open ... 8
opt ... 3, 8
otherForm (tag) ... 47

parents (tag) ... 6, 35
part (tag) ... 6, 10, 36
pe ... 11
ptr (tag) ... 6, 10-11, 37

rec ... 3, 8
remarks (tag) ... 6, 8, 10-11, 39
rend (attribute) ... 32
req ... 3, 8
rs (tag) ... 6, 8, 10-11, 40
rwa ... 3, 8

semi ... 8
specialPara ... 21
string (tag) ... 11, 41
system ... 42

tag (tag) ... 4, 42
tagDoc (tag) ... 3, 6-9, 12, 43, 47
target (attribute) ... 39
TEI (attribute) ... 3
TEI.2 (tag) ... 32
teitsd2.dtd ... 4, 12
term (tag) ... 45-47
termEntry (tag) ... 46
tsd (tag) ... 3, 12, 47
type (attribute) ... 8, 10-11

usage (attribute) ... 3, 8

val (tag) ... 4, 8, 47
valDesc (tag) ... 8, 48
valList (tag) ... 8, 49

xref (tag) ... 11

yes ... 3