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 21
Text Encoding Initiative
Chicago, Oxford

July 15, 1992

Draft Version 2, July 15, 1992

Note

This is the current draft (as of July 15, 1992) 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 1992.

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.

This document is made possible in part by financial support from the U.S. National Endowment for the Humanities, an independent federal agency; Directorate General XIII of the Commission of the European Communities; the Andrew W. Mellon Foundation; and the Social Science and Humanities Research Council of Canada.

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 21

Characters and Character Sets

Computer systems vary greatly in the sets of characters they make available for use in electronic documents; this variety enables users with widely different needs to find computer systems suitable to their work, but it also complicates the interchange of documents among systems; hence the need for a chapter on this topic in these Guidelines.

Four character-set problems arise for the encoder of electronic texts:

1. selecting a character set to use in creating, processing, or storing the electronic text
2. preparing documents for interchange so that the characters within them are not corrupted in transit
3. encoding characters which are not provided by the character set available on the computer system in use
4. signaling shifts from one character set to another, e.g. from the Latin alphabet to Greek and back, or to a special symbol character set and back

This chapter describes the recommended solutions to these problems, in enough detail to satisfy the needs of most users. More detail and more technical information can be found in chapters 61, “TEI Conformance,” [in separate fascicle], 64, “Use of TEI Encoding Scheme in Interchange,” [in separate fascicle], 52, “Writing System Declaration,” [in separate fascicle], and 82, “Standard Writing System Declarations,” [in separate fascicle].

21.1 Local Character Sets

No single character set is prescribed for use in TEI-encoded documents. Users may use any character set available to them, subject to the character set restrictions imposed by the SGML declaration. It is recommended that the character set used be documented by one or more *writing system declarations* (WSD), on which see below. In most cases, a predefined writing system declaration should be suitable. For writing system declarations provided as part of the TEI Guidelines, see chapter 82, “Standard Writing System Declarations,” [in separate fascicle].

In general, it is most convenient to use a character set readily available on one's computer system, though for special purposes it may be preferable to customize the character set using software specialized for the purpose. Whether to use the usual character set or create a custom set depends on the documents being encoded, the staff support and tools available for customizing the character set, the user's technical facility, etc. A choice must be made between the perceived relative convenience of living with an existing character set and the effort of modifying it to suit one's documents more closely. Care should be taken, in particular where multi-byte character sets are used, that the syntactic distinctions required by an SGML processor are preserved. Where a suitable character set is defined by a national or international standard, local customization should implement the standard rather than inventing yet another non-standard character set. The choice must be made by each individual according to individual circumstances; no general recommendations are made here as to whether locally customized character sets should be used. In principle, however, for local processing, encoders should use whatever character set they find convenient.

21.1.1 Characters Available Locally

When the characters in a text exist in the local character set, the appropriate character codes should be used to represent them. Virtually all computer systems provide at least the 52 letters of the standard Latin alphabet, the ten decimal digits, and some basic punctuation marks.

Other characters, such as Latin characters with diacritics (e.g. ä or é) or non-Latin characters (e.g. Greek, Hebrew, Arabic, Cyrillic) are less universally provided. Oriental scripts pose particular problems because of their size. If the local character set provides an *ä* however, there is normally no reason not to use it where that character appears in the text, unless the electronic text is to be moved frequently among machines, in which case one may wish to restrict the electronic text to characters known to translate well among machines. (For more information on moving characters among machines, see section 21.3, “Character Set Problems in Interchange,” on page 9 below.)

As noted above, full use of a local character set may require that the SGML declaration be modified to define all the characters used as legal SGML characters. For further information see chapter 63, “Local Installation and Support of TEI Markup,” [in separate fascicle].

21.1.2 Characters Not Available Locally

Characters not available in the local character set should usually be encoded using SGML *entity references*. In SGML terms, an *entity* (described in more detail in section 12, “Concise Summary of SGML,” [in separate fascicle]) is any named string of characters, from a single character to an entire system file. Entities are included in SGML documents by *entity references*. Lists of suggested names for all the characters and printers' symbols used by most modern European languages have been published by ISO and others.[1]

[1] The most widely used such entity set is to be found in Annex D to ISO 8879; it also reproduced or summarized in most SGML textbooks, notably Charles F. Goldfarb, *The SGML Handbook* (Oxford: Clarendon Press, 1990). A list of some frequently used standard entity names may be found in chapter 82, “Standard Writing System Declarations,” [in separate fascicle]. Extensive entity sets are being developed by the TEI and others are being documented in the fascicles of ISO/TR 9573: *Technical Report: Information processing — SGML support facilities — Techniques for using SGML* ([Geneva]: ISO, 1988 et seq.).

For example, the standard entity name for the character *ä* is *auml*; a reference to the entity gives the name of the entity, preceded by an ampersand and followed by a semicolon.[2] For example, consider the following German sentence:

> Trotz dieser langen Tradition sekundäranalytischer Ansätze wird man die Bilanz tatsächlich durchgeführter sekundäranalytischer Arbeiten aber in wesentlichen Punkten als unbefriedigend empfinden müssen.

Using the standard names for a-umlaut and u-umlaut, one could transcribe this sentence thus:

```
Trotz dieser langen Tradition sekund&auml;ranalytischer
Ans&auml;tze wird man die Bilanz tats&auml;chlich
durchgef&uuml;hrter sekund&auml;ranalytischer Arbeiten
aber in wesentlichen Punkten als unbefriedigend
empfinden m&uuml;ssen.
```

As noted above, standard entity names have been defined for most characters used by languages written in the Latin alphabet, and for some other scripts, including Greek, Cyrillic, Coptic, and the International Phonetic Alphabet. A useful subset of these may be found in chapter 82, “Standard Writing System Declarations,” [in separate fascicle].

Before an entity can be referred to, it must be declared. Standard public entity names can be declared en masse, by including within the DTD subset of the SGML document a reference to the standard public entity which declares them. The German document quoted above, for example, would have the following lines, or their equivalent, in its DTD subset:

```
<!ENTITY % ISOLat1 PUBLIC "ISO 8879-1986//ENTITIES Added Latin 1//EN">
%ISOLat1;
```

Where no standard entity name exists, or where the standard name is felt unsuitable for some reason, the encoder may declare non-standard entities, using the normal SGML syntax. The replacement string for such entities may vary for different purposes, as in the following extended example.

In transcribing a manuscript, it might be desirable to distinguish among three distinct forms of the letter *r*. In the transcript, each of these forms will be encoded by an entity reference, say, “&r;”, “&r2;”, and “&r3;”. Entity declarations must then be provided, within the DTD subset of the SGML document, to define these entities and specify a substitute string.

One possible set of declarations would be as follows:

[2] Strictly speaking the semicolon is not always required; for details see any full treatment of SGML.

```
<!ENTITY r  'r[1]'><!-- most common form of 'r'  -->
<!ENTITY r2 'r[2]'><!-- secondary form of 'r'    -->
<!ENTITY r3 'r[3]'><!-- third form of 'r'        -->
```

The expansions shown above will simply flag each occurrence with a number in brackets to indicate which form of *r* appears in the manuscript.

Another (imaginary) set of declarations might use some printer-specific codes to produce visually distinct versions of the three letters in output:

```
<!ENTITY so '&#27;I&#12;'><!-- shift to alternate printer font -->
<!ENTITY si '&#27;I&#8;'> <!-- shift to basic printer font -->
<!ENTITY r  'r'>           <!-- most common form of 'r'  -->
<!ENTITY r2 '&so;r&si;'>   <!-- secondary form of 'r'    -->
<!ENTITY r3 '&so;&#113;&si;'><!-- third form of 'r'      -->
```

Here the replacement strings for each entity contain *character references*, for example "&#27", indicating the decimal value (or *code point*) of the character to be generated by the SGML processor. The assumption is that the sequence of bytes generated will drive a printer to produce a distinctive form of the letter. Such printer instructions are of their nature highly machine- and software-dependent; by allowing them to be confined to a single place (the entity definitions), SGML makes it easier to adapt such device-specific information to new environments.

Alternatively, the declarations can give all three forms the same appearance in the output:

```
<!ENTITY r  'r'><!-- most common form of 'r'  -->
<!ENTITY r2 'r'><!-- secondary form of 'r'    -->
<!ENTITY r3 'r'><!-- third form of 'r'        -->
```

And finally, to ensure that the SGML output uses the same entity references for these characters as does the SGML input, one could use the following declarations.

```
<!ENTITY amp '&'>
<!ENTITY r  '&amp;r;' ><!-- most common form of 'r'  -->
<!ENTITY r2 '&amp;r2;'><!-- secondary form of 'r'    -->
<!ENTITY r3 '&amp;r3;'><!-- third form of 'r'        -->
```

For transcriptions in scripts not supported by the local character set, entity references may prove unwieldy. In such cases, it is also possible to transliterate the material from its original script into the script of the local character set; like a customized local character set, a transliteration scheme should be documented with a writing system declaration. To avoid information loss, a *reversible* transliteration scheme (i.e. one in which it is possible to reconstruct the original writing from the transliteration) should be preferred; wherever possible, standard schemes should be preferred to ad hoc schemes.[3]
&@fn@sep

[3] Reversible transliteration schemes are defined by national and international standards too numerous to list here; another useful source is *ALA-LC Romani-*

For example, using the Beta code transcription developed for ancient Greek by the Thesaurus Linguae Graecae,[4] one would transcribe the start of the *Iliad* of Homer thus:

```
*MH=NIN A)/EIDE QEA\ *PHLHI+A/DEW *)AXILH=OS
OU)LOME/NHN, H(\ MURI/' *)AXAIOI=S A)/LGE' E)/QHKE,
```

Some standard transliteration schemes for commonly encoded languages are included in chapter 82, “Standard Writing System Declarations,” [in separate fascicle].

21.2 Shifting Among Character Sets

Many documents contain material from more than one language: loan words, quotations from foreign languages, etc. Since languages use a variety of *writing systems*, which in turn use a variety of *character repertoires*, shifts in language frequently go hand in hand with shifts in character repertoire and writing system. Since language change is frequently of importance in processing a document, even when no character set shift is needed, the encoding scheme defined here provides a global attribute *lang* to make it possible to mark language shifts explicitly. This attribute is also used to trigger character-set shifting by application programs.

Some languages use more than one writing system. For example, some Slavic languages may be written either in the Latin or in the Cyrillic alphabet; some Turkic languages in Cyrillic, Latin, or Arabic script. In such cases, each writing system must be treated separately, as a separate “language”. Each distinct value of the *lang* attribute, therefore, represents a single natural language, a single writing system, and a single coded character set.

It is strongly recommended that each value used for the *lang* attribute correspond to a writing system declaration suitable for the character set and writing system being used. The values should where possible be taken from the standard two- and three-letter language codes defined by the international standard ISO 639.[5] When more than one writing system is used for the same language in

---

zation Tables: Transliteration Schemes for Non-Roman Scripts, approved by the Library of Congress and the American Library Association, tables compiled and edited by Randall K. Barry (Washington: Library of Congress, 1991). Many of these schemes, however, use diacritics which are themselves not always available in standard electronic character sets and thus may require careful adaptation for use in electronic work.

[4] Thesaurus Linguae Graecae, *Beta Manual* (Irvine: TLG, [1988]). See also Luci Berkowitz and Karl A. Squitier, *Thesaurus Linguae Graecae Canon of Greek Authors and Works* 2nd edition (Oxford: Oxford University Press, 1986).

[5] ISO 639: 1988, *Code for the representation of names of languages* ([Geneva]: International Organization for Standardization, 1988). A new version of this standard is in preparation which will supply three-letter codes as well as the current system of two-letter codes. Either may be used in TEI documents.

the same document, suffixes should be added to the values from ISO 639. A selection of standard language codes, as well as a number of standard writing system declarations, are provided in chapter 82, “Standard Writing System Declarations,” [in separate fascicle]; other writing system declarations may be provided locally.

Like any global attribute, the *lang* attribute may be used on any element in the SGML document. To mark a technical term, for example, as being in a particular language, one may simply specify the appropriate language on the **<term>** tag (for which see 23.8, “Terms, Cited Words, and Glosses,” [in separate fascicle]):

```
<p lang=EN> ...
But then only will there be good ground of hope for the
further advance of knowledge, when there shall be received
and gathered together into natural history a variety of
experiments, which are of no use in themselves, but simply
serve to discover causes and axioms, which I call
<term lang=LA>Experimenta lucifera</term>, experiments of
<term>light</term>, to distinguish them from those which I
call <term lang=LA>fructifera</term>, experiments of
<term>fruit</term>.
<p>Now experiments of this kind have one admirable
property and condition:  they never miss or fail. ...
```

The form in which materials in different writing systems are processed or displayed is not specified by these Guidelines; if appropriate characters are not available locally, application software may choose to display the material in an appropriate transliteration, as a series of entity references, or in other forms. If the local system requires explicit escape sequences or locking-shift control functions to signal shifts to alternate character sets, these escape sequences may be embedded directly in the content of the appropriate element or may be supplied by the application software upon recognition of the language shift. However, escape sequences are vulnerable to loss or misinterpretation when documents are sent from site to site. For this reason, the method of embedding them directly in the document, while allowed within TEI-conformant interchange, is not recommended for general use.[6]

[6] Standard methods for character-set shifting are defined in ISO 2022: 1986, *Information processing — ISO 7-bit and 8-bit coded character sets — Code extension techniques*, 3d ed. ([Geneva]: International Organization for Standardization, 1986). A standard set of control functions, including methods of specifying script direction (left-right, right-left, top-down, etc.), is defined by ISO 6429: 1992, *Information processing — Control functions for 7-bit and 8-bit coded character sets*, ([Geneva: ISO, 1992). These and related standards are recommended to the notice of those designing systems to handle multiple character sets.

21.3 Character Set Problems in Interchange

Electronic texts may be exchanged over electronic networks, through exchange of magnetic media (e.g. disk or tape), or by other means. In every case except the transmission of storage media from one machine to another machine of the same hardware type running the same operating system and using the same coded character set, the characters are subject to translation and interpretation, and hence to misinterpretation and corruption, by utility software working somewhere on the interchange path. Network gateways, tape-reading software, and disk utilities routinely translate from one character set to another before passing the data on. If the utility errs in identifying the character set, or if several utilities translate back and forth among character sets using non-reversible translations, the chances are good that characters will be garbled and information lost.

At this time (1992), the characters least susceptible to loss or misinterpretation in transit among systems are those shown below, which represent a subset of the characters in the international standard ISO 646 (International Reference Version)[7] and may thus be called the *ISO 646 subset*.

```
a b c d e f g h i j k l m n o p q r s t u v w x y z
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
0 1 2 3 4 5 6 7 8 9
" % & ' ( ) * + , - . / : ; < = > ? _
```

In interchange over any transmission link, the transmitted document should contain only those characters which safely survive transmission over the link; others should be represented with entity references, or with transliterations, as described above.

In blind interchange by means of magnetic or optical media, it is recommended that the document be encoded using some well documented and widely used standard character set.

In blind interchange over networks, it is recommended that the transmitted document contain only characters known to travel safely over the networks involved. In the most general case, those characters are the ISO 646 subset given above.

21.4 The Writing System Declaration

Each language and writing system used in a document should be documented in a Writing System Declaration (WSD), which specifies:

- a formal name for the writing system and language, for use as a *lang* attribute value on elements encoded in it

[7] ISO (International Organization for Standardization). *ISO 646: 1991. Information processing — ISO 7-bit coded character set for information interchange.* [Geneva]: International Organization for Standardization, 1991.

- a specification for the meaning of each character available in the writing system

The characters available within a WSD may be specified in one or more of the following three ways:

- by reference to an international, national, or TEI-registered coded character set
- by reference to such a standard followed by formal declaration of all exceptions
- by providing a formal declaration for each character used

Individual characters within a WSD are formally declared, where necessary, by providing the following information:

- the unique code used to represent the character
- special properties such as whether the character is a diacritic mark or not
- brief textual description of the character
- standard or local entity name used for the character in interchange
- other standard identifiers for the character, if available, such as its code in the "Universal Character Set" of ISO 10646 or Unicode[8] or its standard number in the Association for Font Information Interchange registry.
- optionally, some specification of a suitable graphic rendition for the character in a suitable notation (e.g. graphic image, Metafont program, etc.)

The writing system declaration is one of a set of *auxiliary* documents which provide documentation relevant to the processing of TEI texts. Auxiliary documents are themselves SGML documents, for which document type declarations are provided. The DTD for the Writing System Declaration is discussed in detail in chapter 52, "Writing System Declaration," [in separate fascicle]. Standard Writing System Declarations are provided in chapter 82, "Standard Writing System Declarations," [in separate fascicle].

---

[8] At the time this chapter was published the most recent available version of ISO 10646 was ISO (International Organization for Standardization), and IEC (International Electrotechnical Commission), *ISO/IEC DIS 10646-1.2: 1992. Information technology — Universal Multiple-Octet Coded Character Set (UCS) — Part 1: Architecture and Basic Multilingual Plane.* [Geneva]: International Organization for Standardization, 1992. The most recent published version of Unicode was The Unicode Consortium, *The Unicode Standard: Worldwide Character Encoding* Version 1.0, Volume 1 (Reading, Mass.: Addison-Wesley, 1991). Both of these versions have been superseded by subsequent events: Unicode 1.0 by the ISO DIS 10646, which represents a merger of Unicode 1.0 with an earlier ISO draft; ISO DIS 10646 by the revisions being made in the process of resolving technical comments on the DIS. These changes will be reflected in the official published standard, and in version 1.1 of Unicode.

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, P.O. Box 6998, Chicago IL 60680, USA. Bitnet: U35395@UICVM, Internet: U35395 @ uicvm.cc.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

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!