30 Rules for Interchange

This chapter discusses issues related almost exclusively to the use of SGML-encoded TEI documents in interchange. XML-encoded TEI documents may be safely interchanged without formality over current networks, largely without concern for any of the issues discussed here. This chapter has not therefore been revised, and will probably be withdrawn or substantially modified at the next release.

This chapter describes how interested parties can determine and agree on the proper format for the successful interchange of TEI-conformant documents over a given communications link, and how to translate from normal TEI form to the transmission format and back. It also includes recommendations for formats to be used when private arrangements cannot be made, in non-negotiated or 'blind' interchange.

30.1 Negotiated Interchange

When the sender and receiver of a given text know each other's identity and can make appropriate special arrangements for the interchange of the text, the following procedures may be used to ensure the successful interchange of the text without information loss.

The sender and receiver together must first:

1. agree on whether to exchange the document in *TEI interchange form* or in some other form (e.g. *TEI local-processing form*)
2. identify the *communications link*: the method to be used to transmit and receive the document (transmission over a given network, physical transmission of a disk, tape, or other medium, etc.)
3. identify (by experimentation if necessary) the set of characters which can be transmitted successfully, without corruption, over the communications link; this is the *transmission character set*.
4. identify the set of characters present in the document which lie outside the transmission character set; this is the set of *non-transmissible characters*. For each character in this set, the local writing system declaration should identify an entity name or a transliteration into the transmission character set, which is to be used to transmit the character. The set of entities needed for this purpose is the *transmission entity set*.

The transmission character set is defined as the set of characters in the sender's system character set(s) which survive transmission and are properly recognized in the recipient's system character set. It is therefore by definition a subset of both the sender's and the recipient's system character sets. The bit patterns used to represent the characters may differ in the two systems (e.g. one may use ASCII, the other EBCDIC) if the communications link performs the proper translations.

Current network standards allow — indeed, require — gateway nodes to translate material passing through the gateway from one coded character set into another, when the networks joined by the gateway use different coded character sets. Since there is no universally satisfactory translation among all coded character sets in common use, the transmission character set will normally be the subset which is satisfactorily translated by the gateways encountered in transit between the sender and the receiver of the data.

When material is transmitted on a physical storage medium (e.g. disk or tape), then those exchanging documents have far greater control over the data. If both partners use compatible systems, the transmission character set may be equivalent to their system character sets; otherwise, the transmission character set will include those characters which the recipient can successfully read into the local system character set from the media provided by the sender. For example, if a diskette created by an MS-DOS machine can be mailed to a Macintosh user and read directly by the recipient's system, then the transmission character set is likely to be ISO 646 IRV (equivalent to ANSI X3.4, or ASCII), which both machines have in common; if the recipient's disk-reading utilities are more sophisticated, however, then it may be possible to include some or all of the two machines' non-standard extended characters as well.

The mapping from non-transmissible characters to the transmission entity set may be derived from the writing system declarations in use by the sender and receiver.

After the transmission character set and entity character sets have been defined, the sender must prepare and transmit the document:

1. if the document is to be exchanged in TEI interchange format, translate it from the local-processing form into the TEI interchange form (e.g. by passing SGML through an SGML normalizer to supply omitted SGML tags and interpret all short-reference sequences; XML documents are required to obey these same constraints, and are considered to be in the interchange format)
2. in the case of SGML, pack the document for transmission by replacing every non-transmissible character with the appropriate transliteration or entity reference. At this point, every character in the document is represented either by a character in the transmission character set or by a reference to an entity in the transmission entity set; the document is in *TEI packed format*. If the transmission character set includes all the required delimiter characters, the document will be a conforming SGML document; otherwise, it may not be SGML-conformant but will be easily mappable to a conforming document. In the case of XML, which is always in Unicode, the document may be packed to UTF-8 for transmission.
3. transmit the document over the agreed link

The translation of non-transmissible characters into the transmission entity set may be accomplished under automatic control, by software which reads the appropriate local writing system declarations, creates the necessary mapping tables, and packs the document for transmission.

Upon receipt, the receiver must:

1. unpack the document by expanding those entities which correspond to characters in the local system character set(s); at this point, the document should once more be conformant, and should probably be validated to detect any problems in transmission
2. optionally, material transliterated in the document as transmitted may be translated into the local system character set(s) for more convenient display, or from the transmission entity set into a local transliteration scheme; in XML this should not typically be necessary

It is strongly recommended that when documents are interchanged they be accompanied by any writing system declarations and feature system declarations which are applicable. In TEI-conformant interchange, it is required that documents be accompanied by any applicable tag-set documentation files.

30.2 Some Simple Examples

As a first simple example, consider an SGML document containing English, French, and German, to be transmitted from an IBM-compatible personal computer to a Macintosh, over a long-distance network connection. Uploading test files from the PC to the sender's local network node, sending the file via the network, and downloading the document to the Macintosh, reveal (let us assume) that while all the characters of ISO 646 IRV survive intact, the accented characters of French and German do not survive transmission. In this case, the transmission character set is composed of all the characters of 7-bit ISO 646. The entity set required to handle the non-transmissible characters will include the following (assuming they actually occur in the document):

- agrave
- ccedille
- eacute
- egrave
- ocirc
- oelig
- auml, Auml
- ouml, Ouml
- uuml, Uuml
- szlig

When 'packing' the file for transmission, the sender must replace the non-transmissible characters in the document with references to these entities. After this subsitution, the document is a conforming document written entirely in the transmission character set, which can be sent over the communications link without any garbling or loss of information. Upon receipt, the recipient can replace the entity references with the specific coded characters used on the Macintosh to write French and German.

As a second example, consider the same document being transmitted from a VAX running VMS to an IBM mainframe running VM/CMS. Here, the accented characters might be represented on the VAX using the coded character set ISO 8859-1, but since ISO 8859-1 is not always supported, it may be more likely that entity references will be used instead. Let us assume that the network path between the two machines accepts Latin characters and digits, and most punctuation, but garbles square brackets, braces, the hash mark, and the pounds-sterling symbol. In this case, the accented characters require no special work by the sender, since they are already in a network-safe form. Square brackets, etc., must however be replaced by entity references to lbr, rbr, etc. After this is done, the document is no longer conformant SGML, since the square brackets used in certain markup declarations will not be recognized. (It is important, therefore, for validation to be performed before the square brackets are replaced by entity references.)

Upon receipt, the document may be translated into a valid SGML document by replacing all references to lbr and rbr with the appropriate square brackets, etc. If the local system supports one of the IBM code pages with support for French and German characters, then the entity references to those characters may be replaced by the characters in the system character set. More commonly, the entities for French and German characters will be left in place.

As a third example, consider a document containing Greek, as well as Latin, German, French, and English. If the sender's system has a full Greek character set, but the recipient's does not, then the Greek characters must all be replaced either by references to entities or by transliterations into Latin characters (e.g. using the beta code transliteration developed for the Thesaurus Linguæ Græcæ). If the text is later transmitted to another system which does have a full Greek character set, the transliterated text or entity references may be translated, under control of the relevant writing system declarations, into the local Greek character set.

As a final example, consider a document written in Japanese, to be transmitted over a network within Japan, or over an international network to a recipient in Europe. Since networks within Japan transmit Japanese text without information loss, and common utilities may be used to recognize any of the existing coded character sets and translate into another, the transmission character set for interchange within Japan may be the same as the system character set. (If user-defined extensions are defined, in order to allow the encoding of kanji not present in the standard character sets, then these non-standard kanji may need to be replaced either by entity references or by 'transliterations' into the standard character sets, and the description of the kanji themselves should accompany the documents in which they are used; the writing system declaration may be used for this purpose.)

When transmitting Japanese text outside Japan, the limitations of the networks at the time of transmission must be taken into account; it may be necessary to transliterate the text, or to replace non-transmissible characters with entity references.

30.3 Non-Negotiated Interchange

In some cases, no negotiation between interchange partners is possible, because they do not know each other's identities. Since it is impossible to discover an appropriate transmission entity set by experiment, such interchange requires the use of extremely conservative assumptions about the frailties of network gateways.

Specifically, it is recommended that for non-negotiated interchange the following practices be adopted:

- The transmission character set should be the International Reference Version of ISO 646 (commonly known as ASCII), or Unicode.

- The transmission entity set should include only entities documented in ISO standard entity sets, published TEI writing system declarations, or writing system declarations made available with the material.
- All applicable writing system declarations should be distributed together with the material they describe; in non-negotiated interchange, all writing system declarations should assume ISO 646 (IRV) or Unicode as the system character set.
- For SGML, the SGML declaration distributed with the material should assume ISO 646 (IRV) as the system character set.
- If the receiver is not using ISO 646 IRV or Unicode as the system character set, then the receiver (or some intervening network node) must translate from ISO 646 into the receiver's system character set. For SGML, the receiver or the receiver's unpacking software is responsible for rewriting those parts of the SGML declaration which are dependent on the coded character set, and ensuring that they work properly on the receiving system.
- As transmitted, the document should be valid.

By these restrictions, these recommendations ensure that documents interchanged in this way will be directly usable on a great variety of systems; moreover, since the allowed character sets are widely known and well documented, users of other systems will normally be able to adjust the documentation and data stream to their local systems without difficulty.

It should be noted that ISO 646 imposes a very restrictive and cumbersome encoding for researchers whose character sets have a large repertoire; it is strongly recommended, therefore, that such materials use XML and Unicode, or that arrangements for their interchange involve explicitly negotiated interchange formats wherever possible.

The rules given here for non-negotiated interchange are not guaranteed to succeed, and negotiation of interchange formats is therefore *required*, if any of the following apply:

- The communications link between the interchange partners garbles or corrupts any characters of ISO 646 (IRV), (i.e., ISO 646 (IRV) is not a subset of the transmission character set).
- The communications link maps characters not in the repertoire of ISO 646 (IRV) onto characters in that repertoire.
- The document uses a non-standard SGML concrete syntax (XML does not permit variant concrete syntaxes, so this case does not apply when using XML).

30.4 Notes for Implementors

The descriptions of document interchange in this chapter from time to time refer to software used to pack documents for interchange, or to unpack documents upon receipt. The descriptions do not characterize any specific existing software, but attempt to make clear how such software must work, in a general way. It is hoped that the descriptions will be useful to implementors of packing and unpacking software, but the full specification of such packing and unpacking software is beyond the scope of this chapter. All that can be attempted here is to describe some complications which may arise in the packing and unpacking of documents for interchange, of which implementors of such software should be aware. Most of these difficulties do not arise with XML, because XML requires that the character set be Unicode.

- if the sender and receiver use different SGML syntaxes, various incompatibilities may be encountered which will require one partner or the other to modify the SGML syntax used for the document, or the document itself, or both. In general, unless other arrangements are made, the responsibility for such modifications falls upon the receiver of the document.
- in particular, if the SGML syntax has been modified to expand the set of legal name characters (e.g. to allow characters with diacritic marks to occur in SGML names), then either the recipient must similarly modify the local SGML declaration, or the SGML names must be modified (by sender or receiver) to make them legal under the recipient's SGML declaration; name collisions must be carefully avoided when this is done.

- if the transmission character set does not include all characters with special meaning to the parser (name characters, delimiters, etc.), then although the packed document will be in a one-to-one relationship with a conforming document, it will not itself be conforming. In this case, validation must be done by the sender before packing, and the recipient cannot validate the received document before unpacking it.
- the proper packing of characters for transport may vary from language to language: in English text, a left square bracket may need to be packed with an entity reference to lbr, while the same bracket may have a different meaning, and thus a different entity replacement, in Greek text. In general, this means the packing should be done by an application able to detect element boundaries, read the value of the lang attribute from the start-tag or infer it from context, and adjust its actions appropriately. If only one WSD is in use in a given document and no language shifts are present, then both packing and unpacking may be done by a much simpler string-replacement algorithm.