Background and context
for
the development of a Corpus Encoding Standard

Invitation draft
for comments, revisions and additions

Nancy Ide & Jean Véronis

LABORATOIRE PAROLE ET LANGAGE
U.R.A. 261 CNRS
Université de Provence
29, Avenue Robert Schuman
13621 Aix-en-Provence Cedex 1 (France)

e-mail: `ide,veronis@grtc.cnrs-mrs.fr`

EAGLES - Corpus sub-group on Text Representation
27 September 1993

Contents

Invitation draft for comments, revisions and additions

1. Introduction

The language engineering community has recently revived its interest in the use of empirical methods,[1] thus creating a demand for large-scale corpora. Numerous data-gathering efforts exist on both sides of the Atlantic to provide wide-spread access to both mono- and bi-lingual resources of sufficient size and coverage for data-oriented work, including the ACL Data Collection Initative (ACL/DCI), the European Corpus Initiative (ECI), ICAME, the British National Corpus (BNC), the Linguistic Data Consortium (LDC), etc. The rapid multiplication of such efforts has made it critical for the language engineering community to create a set of standards for encoding corpora.

The goal of the EAGLES corpus sub-group on Text Representation is to develop a Corpus Encoding Standard (CES) optimally suited for use in language engineering, which can serve as a widely accepted set of encoding standards for European corpus work. The overall goal is the identification of a minimal encoding level that corpora must achieve to be considered standardized in terms of descriptive representation (marking of structural and typographic information) as well as general architecture (so as to be maximally suited for use in a text database). It is also intended to provide other levels of more extensive encoding specifications depending on different types of use and applications.

However, the development of such an encoding standard is an enormous task. Although the Text Encoding Initiative (TEI) has taken the first steps, in large part the major issues, concerns, and theoretical basis of text encoding have not yet been established, and have in fact only begun to be adequately and appropriately addressed. This paper attempts to lay the ground for the work of the corpus sub-group on text representation by identifying the issues and concerns which must be addressed before encoding standards for corpora can be developed.

2. Scope of the Text Representation Subgroup

2.1. Text types

The term *corpus* typically designates a collection of linguistic data, including written, spoken, or both, in one or multiple languages. In some cases, the term *corpus* (as opposed to terms such as *collection, archive,* etc.) is further restricted to apply to collections constructed according to various linguistic criteria such as *representativeness* and *balance* across a given domain, set of languages, etc. Here, we use the term *corpus* to refer to any collection of linguistic data, whether or not it is selected or structured according to some design criteria. According to this definition, a corpus can potentially contain any text type, including not only prose, newspapers, as well as poetry, drama, etc., but also word lists, dictionaries, etc. One of the tasks of the EAGLES sub-group will be to delimit the scope of the text types to be addressed and assign priorities, in the light of intended applications.

2.2. Languages

The CES will apply to monolingual corpora including texts from a variety of western European languages, as well as multi-lingual corpora and parallel corpora comprising texts in any of these languages. For language-specific elements, in particular linguistic annotation, the CES will be developed with an eye toward extension to cover additional western and eastern European languages.

2.3. Applications

The CES is intended to be used for encoding corpora used as a resource in language engineering, including all areas of natural language processing, machine translation, lexicography, etc. Corpora are used in language engineering to gather real language evidence, both qualitative and quantitative. Qualitative evidence consists of examples which can be used for the construction of computational lexicons, grammars, and multi-lingual lexicons and term banks, for lexicography, etc. Quantitative information consists of statistics which indicate frequent or characteristic uses of language. These statistics can also be used to guide preference-based parsers, assist in lexicography, determine translation equivalents, etc. In addition, statistics can be used to drive morphological taggers, POS taggers, alignment programs, sense taggers, etc. Common operations on corpora for the purposes of language engineering include extraction of sub-corpora; sophisticated search and retrieval, including collocation extraction, concordance generation, generation of lists of linguistic elements, etc.; and the determination of statistics such as frequency information, averages, mutual information scores, etc.

We do not address corpora intended for other purposes, such as stylistic studies, socio-linguistics, historical studies, information retrieval, etc., although these uses are not excluded *a priori* (in fact, many of the features required for these applications may be the same as those needed for language engineering). Treating a restricted domain enables development of a standard tighter than that of the TEI, by providing specific encoding solutions rather than general or multiple ones, and, most importantly, by providing standards for elements particularly important in that domain (e.g., linguistic annotation).

2.4. Encoded facts

The CES will cover those areas of corpus encoding on which there exists consensus among the language engineering community, or on which consensus can be easily achieved (for example, text divisions, POS tags). Areas where no consensus can be reached (for example, sense tagging) will be identified and, where possible, alternatives outlined. The CES will consist of recommendations for various levels of encoding, corresponding to increasing enhancement in the amount of encoded information, the lowest of which will constitute a minimum level of encoding required to make the corpus (re)usable across all possible language engineering applications. Beyond this, the CES will provide a mapping between each language engineering application and the minimum requirements for that application.

Note that standarization of tags for linguistic information such as part-of-speech and and morphological category is covered by the subgroup on morphosyntactic annotation. The Text Representation subgroup will use the categories developed by this subgroup and provide encoding mechanisms to implement their tagging.

3. Relation to the TEI

The TEI Guidelines [Sper93] (hereafter referred to as TEI-P3) obviously provide a starting point for the development of a corpus encoding standard. However, the TEI scheme will need careful examination, evaluation, and adaptation for corpus encoding. The TEI scheme is largely untested on corpora, especially multi-lingual corpora, and use of the TEI scheme for corpus encoding will therefore almost certainly require modification and extension in the light of experience on real-scale data, for instance, to handle multi-lingual text alignment and alignment of different levels of speech representation (signal, orthographic transcription, phonemic transcription, prosody). In addition, the TEI scheme is not complete, and many areas are yet to be addressed; there is, for example, no TEI encoding scheme for newspapers, some aspects of spoken materials, such as prosody (F0 modelling, symbolic coding, etc.), etc.

More importantly, there are requirements for a corpus encoding standard that the TEI scheme does not, and is not intended to, answer. First, the TEI scheme is intended to be maximally applicable across a wide range of disciplines, including not only corpus linguistics, but also other areas of computational linguistics, the humanities and social sciences, publishing, library science, etc. As such the TEI Guidelines offer solutions for encoding a variety of textual facts, but do not recommend which facts are to be encoded in a corpus. It is therefore necessary to identify the subset(s) of TEI elements that are relevant for corpus-based research according to the intended application and/or use (e.g., interchange only, search and retrieval, validation, etc.).

Because it aims at maximal generality, the TEI necessarily takes its encoding solutions to the highest possible level of abstraction. In addition, the TEI often provides multiple options for encoding the same phenomenon. The need to provide mechanisms which are maximally general and flexible is at times at odds with the provision of mechanisms which are most efficient and/or effective for a specific application or intended use. To develop an encoding standard specifically suited to corpora, it is necessary to choose from among various encoding options the method that is optimal for corpus-based research, in the light of intended use. It may also be advantageous to refine or delimit TEI solutions which are over-general for the needs of corpus encoding.

Finally, and perhaps most importantly, it is outside the scope of the TEI to provide recommendations for certain content-related elements. For example, while the TEI provides several means to *mark* POS, it is not within the scope of the TEI to provide a standardized set of POS category *names*. Instead, it provides a flexible mechanism that can accomodate any set of actual POS category names. Similarly, the TEI does not provide recommendations for names which might, for example, be used as identifiers for texts or text categories, nor does it provide text typologies, categories, design criteria, etc., which are required in order to ensure standardized documentation procedures for corpora. This level of specification would also be required in a comprehensive corpus encoding standard.

We can identify several possible levels of text standardization, corresponding to increasing levels of specificity in markup, in terms of both the kind of features that are marked and their relevance for particular application domains. These levels are defined in the following section. Development of the CES will involve taking standardization to levels beyond those covered by the TEI, as well as identifying the levels to which standardization should be taken for minimum conformance, etc.

The CES will be built upon the general text encoding guidelines developed by the TEI. Development of the standard will involve refining and extending the TEI Guidelines to both suit the specific goals of corpus-based research and provide standards for encoding elements and features beyond the scope of the TEI. At the same time, this will enable evaluation of the TEI Guidelines and proposals for their revision and extension.

4. Levels of standardization

We distinguish three levels of text standardization. The successive levels are increasingly prescriptive in terms of the encoding conventions that must be used to conform to that level of standardization. Each level requires standardization at the preceding level as a prior condition. In addition, the three levels of standardization are interdependent; that is, decisions at one level will affect what can be done at the next level.

Because each of the three levels imposes increasing uniformity of encoding, the data become more and more reusable as standardization is tightened. At the same time, the application areas within which the data are reusable typically become more and more restricted. Thus, there is a trade-off between generality and reusability as the level of standardization is increased.

4.1. Metalanguage level

An encoding standard at the metalanguage level defines a markup *syntax* and the basic *mechanisms* of the markup scheme. The markup syntax specifies the form of tags, the form and maximum allowable length of identifiers, character sets, etc. It does *not* specify the markup itself (tag names, allowable sequences of tags, etc.). The mechanisms include the means to define tags and the structural relations among them, etc. Apart from what is defined at this level, no character in an encoded text has any semantics other than its identity as a graphic character.

The Standard Generalized Markup Language (SGML), with the reference concrete syntax formally defined in ISO 8879 (see [Pric94], [Gold90], [Brya88], and [Herw91]), is unique in that it is a standard at the metalanguage level *only.* That is, the SGML reference concrete syntax defines tag form (any sequence of characters surrounded by "<" and ">"), the base character set, naming rules, reserved words, allowable features (e.g., omission of end tags), etc., but not actual tag names or rules for their use in a document.[2] Using the SGML Document Type Definition (DTD) mechanism, the user can define tag names and document models which specify the relations among tags (this constitutes the markup specification level--see below), utilizing the conventions laid out in the reference concrete syntax.

SGML also enables users to specify their own concrete syntax if desired. For the metalanguage level part of its standard, the TEI uses a somewhat modified version of the SGML reference concrete syntax--for example, the TEI standard[3] extends the allowable length of identifiers and does not allow omission of end tags, but uses the same tag delimeter syntax, etc. However, for the purposes of encoding the complexity and wide range of many of the texts treated by the TEI, it was necessary for the TEI to considerably extend its metalanguage level specification beyond what is offered by SGML. For instance, the TEI provides additional mechanisms for

- defining the meaning of characters (the Writing System Declaration--see the chapter entitled "Writing System Declaration" in TEI-P3)
- defining well-formed feature structures (Feature System Declaration--see the chapter entitled "Feature System Declaration" in TEI-P3)
- specifying complex intra- and inter-textual references (see the chapter entitled "Additional Tag Set for Segmentation and Alignment" in TEI-P3)
- building DTDs from a modular set of pre-defined fragments and a class system for tagsets (see the chapter entitled "The Structure of TEI Document Type Declarations" in TEI-P3).

4.2. Markup specification level

Standardization at the metalanguage level does not fully achieve the goal of universal document interchange, since it is possible, for example, to have entirely different document structures and markup even though the documents are encoded using the same metalanguage level specifications (in SGML, this would mean that the documents have different DTDs). A more powerful way to standardize texts is not only to specify tag syntax, naming rules, etc., but also to identify a relevant set of text categories and specify the actual tags that are to mark these categories in the text together with rules for using these tags. Most familiar markup schemes are at this level: they provide precise tag names and rules for using the tags (i.e., the context(s) in which they can legally appear). In SGML terms, standardization at this level means that documents of the same type have common DTDs. The largest part of the TEI Guidelines constitutes a standard at this level.

This level specifies not only a precise syntax for each tag and for well-formed tag sequences, but it also specifies a semantics for each tag. However, the semantics associated with tags differs depending upon whether the markup is *procedural* or *descriptive* [Coom87]. Procedural (or prescriptive) markup, such as that in typographic markup systems, defines a tag by associating a processing method with it (for example, the tag **.BO** signals that the following text is to be processed by being printed in bold-face type. Descriptive markup does not (necessarily) associate a process with a given tag. Rather, it describes the kind of information that should appear within the tag, independent of any processing that may be applied. Thus a purely descriptive markup scheme, such as the TEI, is application-independent since processing possibilities remain open.

It is important to note that in a descriptive markup scheme, a semantics is also associated with each tag, in terms of the kind of information that the tag is intended to mark. However, these semantics are informal rather than formalized in a precise procedure as is the case for prescriptive markup. For example, a tag such as **<title>** is likely to be used to mark those things which humans more or less agree upon to be titles. This kind of semantics is typically specified in accompanying user manuals; TEI-P3 is an extensive example of the specification of tag semantics at this level.

The TEI specifically provides tag semantics at this level only. In particular, the TEI defines tags corresponding to commonly accepted conceptual categories (such as *title, citation, word,* etc.) and relies on human judgement to apply the category appropriately *according to the user's needs and the intended use.* Detailed semantics, upon which there may be disagreement (for example, "SGML: An Author's Guide" could be seen as a title or a title and a subtitle--the TEI provides ways to mark it either way but does not prescribe the interpretation), are expressly excluded from the TEI.[4] Similarly, the TEI provides a tag for marking words (**<w>**) but leaves it to the user's judgement to provide the definition of the object to which it is to be applied (e.g., orthographic words only as opposed to conceptual units such as compound words).

4.3. Markup use level

Standardization at the levels described above is not enough to make data directly and immediately reusable in many contexts, in particular, within well-delimited domains such as NLP applications. Choices among encoding options and refinements to the TEI scheme must be made in the light of precise goals and specific processing needs. The EAGLES corpus sub-group for Text Representaion will be largely concerned with the development of corpus encoding standards at this level. Ultimately, it will likely be necessary to develop several analogous sets of inter-dependent, customized text encoding standards, each suited to a specific domain and built upon the foundation of mechanisms and principles provided by the TEI Guidelines.

The Text Representation sub-group will build upon the TEI scheme to specify the following:

1. *Required/recommended/optional elements to be marked.* The TEI provides some minimal requirements and recommendations, but they serve only very general goals (e.g., interchangeability) and are not domain-specific. For particular applications, it is possible to go much farther. For example, in any usable corpus for NLP research, paragraph and sentence markup should be required, and explicit marking of items such as abbreviations should be recommended.

2. *Precise choices among options.* To be flexible, the TEI necessarily provides multiple solutions for the same encoding problems. This opens the door for (potentially substantial) encoding differences and variations in encoding style which may make reusability more problematic than it should be within well-delimited domains. For example, structural divisions in text can be marked using nested **<div>** tags or with **<div1>**, **<div2>**, etc. Also, there are several alternative mechanisms by which POS for each word in a corpus could be tagged (feature structures, entities, etc.), and one method could be chosen which is most usable for use within a given domain.

3. *Extensions to the TEI Guidelines to serve needs of a particular domain.* The TEI Guidelines are not complete, and several areas are yet to be addressed which will demand extension of the scheme. In addition, TEI encoding solutions are, as a rule, taken to the highest level of abstraction in order to make them as widely applicable as possible. For well-delimited domains, some solutions may be unnecessarily over-general, thus losing expressive power and potentially reducing processing capabilities. In other cases certain elements are so commonly marked in a given domain that a more specific tag or mechanism is warranted.

4. *Precise values for elements which currently constitute open lists of attribute values in the TEI, and many kinds of tag content.* It is beyond the scope of the TEI to provide encoding recommendations for a substantial number of open-valued elements (for example, part of speech (POS) categories, text typology nomenclature for use in the TEI header, etc.), which, for maximum reusability, should be standardized for use in particular domains. For example, if a corpus to be used for NLP research is marked for POS, but the POS category system differs from that used by a particular researcher, the corpus is unusable until some (possibly non-trivial) mapping is made between the two systems. In some areas, one or more standards already exist (e.g., ISO country names,

bibliographic style) but the TEI, for the sake of generality, does not adopt a particular standard. In other cases, values are not provided due to a lack of consensus within the user community. For example, there is no consensus on a definitive set of POS categories which can be used across all domains (NLP, linguistic research, etc.); yet it is likely that consensus on POS categories could be achieved for use in a specific application, such as NLP applications involving western European languages.[5]

5. *Detailed semantics for elements relevant to language engineering.* The TEI provides general, domain-independent definitions for most elements, but in precise domains such definitions can often be taken farther. For example, the TEI provides no precise definition of what should be marked as a sentence, leaving it to the user to decide. For a specific domain, it could be useful to define "word" or "sentence" in orthographic terms; for another domain it may be useful to provide a more linguistically-based definition.

5. Criteria for standard design

We outline here a number of criteria which have emerged, often from the work of the TEI, as fundamental to the design of text encoding standards. This list is not exhaustive but is intended to provide a starting point for the development of a CES.

5.1. Completeness

An obvious criterion for a CES is completeness--that is, does it provide the ability to encode those features and properties of texts that are required by users. Completeness is obviously an elusive goal since the categories are open-ended; it is nonetheless possible to achieve *practical completeness* by identifying a set of features for any given text type as well as across text types that will serve a large percentage of corpus encoding needs. One of the goals of the EAGLES sub-group will be to assess which percentage of the possible range of encodable features should be covered. Absolute completeness is impossible; therefore, another criterion for the CES will be extensibility (see section 5.9 below).

The completeness of the TEI Guidelines for encoding corpora remains to be tested, since in many areas there was very little or no existing practice. Note that in some areas, such as newspaper encoding, the TEI does not yet provide any encoding guidelines. The EAGLES sub-group will identify such uncovered areas and determine an appropriate strategy for dealing with them.

5.2. Consistency

A CES should be based on some consistent principles for determining what kind of objects are tags, what kind of object is an attribute, and what kind of object(s) appear as tag content. This task will require considerable attention because practice within the TEI is not specified and some inconsistencies are apparent, for example:

- In some cases, tag names refer to very generic categories and specific descriptions are put in attributes; in other cases, the tag name is specific. For example, for verse, hierarchical divisions are marked as **<div1 type=canto>**, **<div2 type=stanza>**, etc. (see [Chil94]) In dictionaries, divisions are explicitly named with **<entry>**, **<homograph>**, **<sense>** (see [Ide94a]). The reason behind this is that for verse, an attempt was made to accomodate *all* relevant literary genres and cultures, whereas for dictionaries, all but western language dictionaries were explicitly excluded from consideration in order to enable a tighter definition of entry structure. In some other similar cases where consensus would exist in the large majority of cases, the tighter description is not accomodated: for example, although the division of book into chapters seems an obvious candidate, there is no **<chapter>** tag, and it is necessary to use **<div type=chapter>** (or **<div*n* type=chapter>**).

- For the most part, the TEI practice seems to be to include what might be regarded as a part of the text "content" as tag content, and to specify other material as attributes. However, the practice is not fully consistent: for example, in the encoding of spoken texts, certain semi-lexical items which are a part of the transcription content, such as "hmm" or "ha", can be encoded as attributes rather than tag content. Conversely, some material which is not a part of text content, for instance, notes and comments which might be made in the process of encoding the text, are included as tag content, even though they may not be considered to be a part of the text content proper.

- Most of the time tags identify structural or logical pieces of a text (e.g., paragraph, segment, utterance, date, etc.) of a text; sometimes they designate

> properties or processes (e.g., regularized, corrected, highlighted, emphasis, etc.). At other times properties are given in attributes. For example, one can say
>
> ```
> <name reg='Volange, de'>de Volanges</name>
> ```
>
> but not
>
> ```
> <w reg='though'>tho</w>
> ```
>
> (**<w>** marks a word, and "reg" indicates regularization). In the second case it is necessary to say
>
> ```
> <w><orig reg='though'>tho</orig></w>
> ```
>
> which is inconsistent with the first example.
>
> When tags are used to mark not only structural or logical objects in a text, but also to describe their properties, considerable verbosity can result. For example, if it is also necessary to indicate that the word is emphasized (**<emph>**), it is necessary to say
>
> ```
> <w><emph><orig reg='though'>tho</orig></emph></w>
> ```
>
> Note also that in this example any permutation of the three tags **<w>, <emph>,** and **<orig>** is allowed to mark the same fact--leading to meaningless combinatorics. This seriously weakens the structural model of the text, since at a certain level virtually anything can contain anything, thus creating problems for validation, search, etc.

It seems clear that a well-thought out system with strong principles (for example, tags for structural and logical pieces, attributes for properties, etc.) would ensure the intellectual integrity and coherence of the encoding scheme and provide a basis for those who modify or extend the TEI.

A lack of such a principled basis leads to practical problems in processing an encoded text, for example, for validation, search and retrieval, etc., since different encoding styles can be mixed within the same document. For example, there are six different ways to encode the last example above, and potentially all of them could appear in the same document using the TEI DTDs. Similarly, the large number of optional attributes allowed by the TEI leaves the door open for within-document inconsistencies, since an attribute may appear on one instance of a tag and not on the next, etc. This leads to validation problems (see 5.4 below).

5.3. Recoverability

When a text is encoded from a printed or electronic source (typesetter's tapes, etc.) the ability to recover the source text from the encoded version--that is, to distinguish what was in the source from the markup and potential additional information--is often desirable. There are a number of different ways to define what is to be recovered from a source text, (e.g., a facsimile of a particular printed version of a text, layout, typography, etc.). For many purposes (comparison and validation between the source and the encoded text, operations such as word counts, search, concordance generation, linguistic analysis, etc.), it is sufficient to recover the sequence of characters constituting the text, independent of any typographic representation.

Recovery is an algorithmic process and should be kept as simple as possible, since complex algorithms are likely to introduce errors. Therefore, an encoding scheme should be designed around a set of principles intended make recovery possible with simple algorithms. Processes such as tag removal, simple mappings are more straightforward and less error prone than, say, algorithms which require rearranging the sequence of elements, or which are context-dependent, etc. For example, a simple way to recover the original character sequence would be to employ the following principles:[6]

1. None of the original sequence of characters (with the possible exception of rendition text) should be deleted or altered.
2. The original data should not be given in attributes, but should always appear as tag content.
3. Apart from the original data, no other data should appear as tag content.
4. The original order of the data should not be changed.

This is obviously a simplification for the sake of illustration, but it reflects a strategy which, although not explicitly stated as a principle, is followed more or less consistently in the TEI Guidelines. In order to provide a coherent and explicit set of recovery principles, various recovery algorithms and a related encoding principles need to be worked out, taking into account such things as the role and nature of mappings (tags to typography, normalized characters, spellings, etc. with the original, etc.), the encoding of rendition characters and rendition text, definitions and separability of the source and annotation (such as linguistic annotation, notes, etc.), linkage of different views or versions of a text, etc.

One of the tasks of the EAGLES sub-group will be to determine the recoverability needs for language engineering application and provide a precise policy in the light of these needs.

5.4. Validatability

In SGML jargon, *validation* refers to the process by which software checks that the markup in a document conforms to the structural specifications given in the DTD--that is, that tags are properly nested, appear in the correct order, contain all required tags, etc.; that attributes appear when and only when they should, have valid values; etc.[7]

The ability to validate is important because it enables trapping errors during data capture. It also enables ensuring that the encoded text corresponds to the model given in the DTD, thus providing a possible means by which the adequacy of the model itself can be verified.

The TEI encoding solutions are typically taken to the highest level of generality, in order to accomodate a wide range of texts, disciplines, and applications, as well as different academic theories, languages, cultures, and historical timeframes. As a result,

- TEI DTDs are *over-generative*--that is, they place very little restriction on where tags can appear relative to one another, in order to allow for even the most exotic of structures.
- tags are often specified at a high level of abstraction, for example, the general **<div>** tag vs. the more specific **<chapter>**.

There is a tension between the generality of an encoding scheme and the ability to validate. Over-generative DTDs allow many tag sequences which, for any given text, are not valid. For example, to accomodate a wide range of dictionary structures, the

TEI DTD for dictionaries allows most elements--orthographic form, pronunciation, grammatical information, etc.--to appear at any level inside an entry, and in any order. For any given dictionary, the structural rules are almost always much more constrained. For example, pronunciation may appear only at the highest level inside an entry and never within a given sense specfication, and it may appear only following the orthographic form. However, with the TEI DTD it is not possible to validate that the tighter structural rules of this dictionary are followed.

The use of abstract, general tags also constrains the ability to validate. For example, the use of a general tag such as **<div>** to mark hierarchical divisions of a text (corresponding, for example, to book, chapter, section, etc.) disallows constraints on what can appear within a given text division. The **<div>** tag has to be defined to allow titles, paragraphs, etc. It is not possible to ensure that tighter structural constraints for a given book are observed, for example, that titles do not appear within chapters, or that a paragraph does not appear outside the chapter level, etc.[8]

Because of the generality of the TEI scheme, it is likely that users will take extensive advantage of the TEI's mechanisms for modification and extension of the TEI DTDs to develop customized encoding formats in order to achieve better validation. This will lead to an increasing number of TEI "dialects" or sub-schemes.

5.5. Compactness

SGML is often criticized for its verbosity, since document size can be dramatically increased by the addition of SGML tags. This is a particular concern for annotated corpora, where each word (and possibly each morpheme) can be marked for part of speech and/or other information, often increasing file size by a factor of 10 or more. This can cause problems for various kinds of processing (e.g., retrieval) as well as for interchange, since in the state of the art it is still often problematic to transfer large files over data networks.

The Text Representation subgroup should evaluate several possible means to reduce the number of characters added to a text when markup is introduced:

- *tag minimization*: SGML provides many means for minimizing the amount of markup in a text via mechanisms such as start and end tag omission, short start and end-tag, minimization of attribute values, etc. For example, the following definitions[9] allow end tag omission:

```
<!ELEMENT w      - O  (orth, pos,lem) >
<!ELEMENT orth  - O (#PCDATA) >
<!ELEMENT pos   - O (#PCDATA) >
<!ELEMENT lem   - O (#PCDATA) >
```

The following is a full markup for the sentence fragment "The boat sinks...":

```
<s>
<w><orth>The</orth><pos>DET</pos><lem>the</lem></w>
<w><orth>boat</orth><pos>NNS</pos><lem>boat</lem></w>
<w><orth>sinks</orth><pos>VBZ</pos><lem>sink</lem></w>
...
</s>
```

With end tag omission this could be replaced by

```
<s>
<w><orth>The<pos>DET<lem>the
<w><orth>boat<pos>NNS<lem>boat
<w><orth>sinks<pos>VBZ<lem>sink
...
</s>
```

which in this case is a nearly 50% reduction in the number of characters.

- *tag renaming:* the TEI provides modifiable DTDs, in which elements are not referred to directly by their identifiers, but rather by parameter entities. This allows simplified renaming of elements by redefining the appropriate parameter entities. Using this strategy, the above example might be tagged as

```
<s>

<w><o>the<p>DET<l>the

<w><o>boat<p>NNS<l>boat

<w><o>sinks<p>VBZ<l>sink

...

</s>
```

- *SGML entities:* SGML allows string substitution via entity replacement. Entity references can be used in place of any string, possibly including markup. So, for example, a complex feature structure specification which occurs frequently in the text can be replaced by an entity reference consisting of only a few characters. The feature structure

```
  <fs type='word structure' id=vbidprx0sgp3>

     <f name=category><sym value=verb></f>

     <f name=mood><sym value=indic></f>

     <f name=tense><sym value=pres></f>

     <f name=auxiliary><minus></f>

     <f name=agreement>

        <fs type='agreement structure' id=sgp3>

            <f name=number><sym value=sg></f>

            <f name=person><sym value=3></f>

            </fs></f>

     </fs>
```

could be replaced by the entity reference **&VBZ;**. Analogous substitutions for other word categories could yield the following encoding:

```
<s>

<w><orth>the&DET;<lem>the

<w><orth>boat&NNS;<lem>boat

<w><orth>sinks&VBZ;<lem>sink

...

</s>
```

- *DATATAG feature:* When certain tag sequences occur with regularity, it is possible to define a certain character to be interpreted as the end tag of an element. For example, the following declarations specify that the character "|" can be interpreted as the end tag for **<orth>** and **<pos>**:

```
<!ELEMENT w     - O  ([orth,"|"], [pos,"|"], lem)  >

<!ELEMENT orth  O O (#PCDATA)                      >

<!ELEMENT pos   O O (#PCDATA)                      >

<!ELEMENT lem   O O (#PCDATA)                      >
```

<orth>, **<pos>**, and **<lem>** are also defined so as to allow omission of both the start and end tags. This yields the following possible encoding:

```
<s>

<w>the|DET|the

<w>wash|NNS|wash

<w>sinks|VBZ|sink

...

</s>
```

If we also specify that the carriage return implies the end-tag of element **<w>**, the encoding could be reduced even further to

```
<s>

the|DET|the

wash|NNS|wash

sinks|VBZ|sink

...

</s>
```

- *non-SGML notations:* It is also possible to use private, less verbose non-SGML schemes within tags or as attribute values. For example, the encoder could decide to use a private notation within the **<s>** element in the example

> above--if that notation uses the pipe sign as a separator between word, part of speech, and lemma, the encoding would be exactly as given above. However, the DTD would simply specify
>
> ```
> <!ELEMENT s      - - (#PCDATA)                       >
> ```
>
> which means that the SGML parser will not process the content of the **<s>** tag in any way. The content would have to be processed by other software. This is in contrast to the use of DATATAG above, where the SGML parser (assuming the optional feature DATATAG is implemented) will understand and process the content of the **<s>** tag as consisting of three elements.

Each of these methods for markup reduction seems to have drawbacks. For example, renaming is likely to introduce conflicts with pre-existing TEI tags such as **<p>** and **<l>**; the use of entities results in considerable processing overhead; some features such as DATATAG are not implemented in all SGML processors; etc. In addition, the TEI forbids the use of some mechanisms such as start and end tag omission in TEI-conformant documents intended for interchange, and others (such as DATATAG) under any circumstances.

The Text Representation subgroup must evaluate the markup minimization issue in order to

- determine the degree to which and the circumstances under which markup minimization is important for corpus encoding;

- ascertain precisely the advantages and drawbacks of the various minimization methods;

- examine the TEI decisions concerning restrictions on the use of various mechanisms.

5.6. Readability

Another criticism of SGML is that the complexity of the encoding often renders a text unreadable by humans. There are two points of view concerning readability. One assumes that the text will be captured, displayed, or in general dealt with using processing software which could make the markup either invisible or human-readable; therefore, readability need not be a concern. However, it can be argued that such software is not readily available, and further that no software will ever answer all the user's needs. Therefore, there will always be a need for dealing directly with the encoded text.

The subgroup must determine the importance of readability for corpora in the intended EAGLES applications and make appropriate design decisions. These decisions will, of course, have repercussions for other criteria such as compactness, processibility, etc. For instance, some minimization techniques compound the readability problem, for example when tag names are too cryptic to be understandable, or when entities are used indiscriminantly. On the other hand, other minimization techniques, such as end tag omission, often improve readability.

5.7. Capturability

Data capture involves

1. capture of the text itself, either by hand or via OCR, acquisition of word processor output, typesetter tapes, etc.; we assume that by-hand capture is not very likely for EAGLES applications, although it is not excluded.
2. addition of markup. Fully automatic markup is rarely possible; markup is typically achieved either by hand or semi-automatically, via format translators, annotation programs such as POS taggers, etc.

The kind of markup that is added to a text directly affects the costs of capture. Some kinds of markup can be very costly, if, for example, no program can accomplish it automatically or if markup programs leave so many ambiguities that a large amount of post-editing is required. Capturability is an important concern because corpora often consist of millions of words of text, making hand marking and substantial post-editing too costly to be practical.

Capturability therefore has implications for the CES:

1. The design of the scheme itself should accomodate the various levels of analysis of the text, and provide markup for both very crude element designation (which can be much less costly to achieve) as well as more precise tagging. For example, markup indicating that a word or any arbitrary segment appears in italics already exists in many texts (such as typesetter's tapes), and it

is therefore virtually cost-free to mark it as such; to determine more precisely what the italics mean can be much more costly, since italics can indicate any number of things (title, caption, quotation, emphasis, foreign word, term, etc.). Similarly, it may be cheap and sufficient for many applications to make only a gross distinction between the main text (to which one may want to restrict linguistic analysis, e.g.) and auxiliary text (titles, divisions headers, captions, tables, footnotes, bibliographic references, etc.). Therefore, the markup scheme should be refinable, by providing tags at various levels of specificity together with a taxonomy identifying the hierarchical relations among them. For example, a word marked in italics could later be further analyzed and identified as a highlighted word, and later more precisely marked as a term, and still later further identified as a foreign term, etc.

2. Minimum requirements for conformance to the standard (see section 6.2) must be made in view of the costs of capture. Minimum requirements cannot include tagging that is actually or even potentially costly. For example, requiring that italics are disambiguated to the lowest level of the hierarchy would result in high costs for data capture since it requires substantial hand intervention. Even seemingly simple tagging, such as tagging paragraphs, can be costly depending on the input, if, for example, line breaks are not differentiated from paragraph breaks (as in electronic mail, etc.).

5.8. Processability

There are two general uses for a text encoding standard:

- data *interchange* between individuals or sites. A data interchange standard means that it is necessary to translate only between a single format and a local format in order to use an externally-acquired text, as opposed to the pair-wise translation between all possible local formats. It also serves to identify a set of common text categories and text models.
- *local processing*, including data capture, as well as various applications such as text editing and formatting; search and retrieval; linguistic, semantic, metrical, etc. analysis; text collation; etc. A local processing standard enables software compatibility, thus avoiding the reinventing of software with the same functionality to suit individual encoding formats. It also enables using

the same text with programs with different functionalities, without the need to translate it into the (potentially different) encoding format required for each.

The TEI provides guidelines intended primarily for data *interchange* between individuals and research sites.

A standard for local processing would involve, potentially, considerations different from those used to design an interchange standard. An interchange standard must necessarily be domain and application-independent, and therefore maximally general. Ideally, it must be as expressive as any local scheme in order to enable translation of all local formats into the interchange format. It need not, and probably should not, be designed primarily on the basis of processing concerns. A standard for local processing, on the other hand, may take processing demands as a fundamental design principle, or may at least weigh them heavily in making design choices. For instance, if one construct or mechanism demands substantial processing overhead, it might be rejected in favor of a less powerful mechanism, if in fact the added power is seen as too high a price to pay for the overhead it incurs.

Because different criteria apply, the standards for interchange and for local processing may not be identical. We can define four possible relations between the two:

1. The interchange and local formats are identical.

2. The local format is a restricted variant of the interchange format, that is, the local format constitutes some subset of the mechanisms and/or tags of the interchange format. This might be done in order to simplify processing. However, information is potentially lost in translating to the local format, and reconstruction of the interchanged text from the local format may be impossible.

3. The interchange format is a restricted variant of the local format. This is the case within the TEI, which places more restrictions on the use of SGML mechanisms for interchange than for local processing--for example, OMITTAG is valid for local processing but not for interchange. In all cases care has been taken to ensure that the interchange format can be generated from the local format. Otherwise, the potential for information loss mentioned above applies here as well.

4. The local format and interchange formats are mutually mappable, but are completely different from one another. In this case, the two formats do not share a common metalanguage level specifications.

The Text Representation subgroup must decide whether the CES is a standard for interchange, local processing, or both; and if both, which of the relationships defined above is most appropriate. Some TEI decisions in this area should be reassessed, since the situation has changed considerably since these decisions were made, and, in particular, because these decisions may not be most appropriate for users of the CES. For instance, some TEI design decisions are based on the possibility that users of an encoded document will not have access to SGML-aware software (for instance, by forbidding most kinds of tag minimization for interchange), which may not be the case for users of the CES in the mid-1990's.

The Text Representation subgroup must also examine the processing costs of different mechanisms used in the CES. For example, the subgroup should determine the costs of entity replacement (including the numerous parameter entities needed by the TEI modifiable DTDs), various kinds of tag minimization, and the use of optional features such as DATATAG, as well as the influence of the size of the DTD (which is very large for the TEI) on processing costs, etc.

5.9. Extensibility

As mentioned above in section 5.1, absolute completeness of any markup scheme is impossible to achieve. Therefore, it is essential that any encoding scheme be extensible.

The TEI provides several mechanisms to enable extensibility, such as mechanisms to enable easy DTD modification, provision for non-SGML escapes for graphics, etc. However, if users take extensive advantage of the TEI's mechanisms for modification and extension of the Guidelines to develop customized encoding formats, the result will be an increasing number of TEI "dialects" or sub-schemes. Without some strong principles for extension and coordination by the TEI, this could paradoxically result in rebuilding the same kind of encoding Tower of Babel that the TEI was expressly established to avoid.

The TEI provides a modular system for DTD construction by means of which DTDs are built from by combining DTD fragments or "tag sets" in any of a number of ways.[10] User extensions can be easily accomodated by introducing additional modules. The TEI encoding scheme also uses a classification system for elements, based on shared SGML attributes and/or a common location in the contents models of TEI DTDs. Classes may have super- and sub-classes, and the characteristics of a super-class (e.g., associated attributes, ability to appear at a given point in a document) are inherited by all members of its subclasses. These classes are formally defined in SGML through the use of parameter entities, thus making it easy to add new elements to existing classes and rename or "undefine" existing elements without extensive revision of the TEI DTDs.

The TEI modular DTDs and class system for elements provide a starting point, but a more systematic approach to extensibility is needed. For instance, it would be possible to develop taxonomy of DTDs (or text types) based on the level of their generality or applicability, provide mappings among DTDs of differing generality, etc. This would provide a detailed scheme to which, for example, users could define additional subtypes of existing types, etc. The Text Representation subgroup should examine these possibilities and provide mechanisms to enable users of the CES to extend the scheme in a systematic and controlled way.

6. Encoded facts

As outlined above in section 4, the CES will provide a set of standard encoding practices that goes beyond the TEI by providing stricter requirements concerning which facts within a text must be encoded in order for the text to be conformant to the standard. Corpora may be encoded at many vastly different levels of analysis (e.g., gross structural analysis, precise linguistic analysis, etc.); therefore, rather than defining a single set of requirements which must necessarily accomodate the lowest level, the CES will define a minimum set of requirements for corpora encoded at each of several different levels of analysis.

6.1. Levels of analysis

We provide a preliminary definition of different levels of analysis for corpora. A first task of the Text Representation subgroup will be to examine and refine this definition and provide a precise inventory of features included in each level.

We can distinguish four levels of document markup:

Level 0. Document-wide markup. This level provides global information about the text, its content, and its encoding. This level corresponds roughly to the TEI header. For example:

- bibliographic description of the document;
- character sets and entities;
- description of encoding conventions;

etc.

Level 1. Gross structural markup. This level includes universal text elements down to the level of paragraph, which is the smallest unit that can be identified language-independently; for example:

- structural units of text, such as volume, chapter, etc., down to the level of paragraph; also footnotes, titles, headings, tables, figures, etc.;
- features of typography and layout, for previously printed texts: page breaks, list item markers;
- non-text ual information (graphics, etc.).

etc.

Level 2. Markup for sub-paragraph structures. This level explictly marks sub-paragraph structures which are usually signalled (sometimes ambiguously) by typography in the text and which are language dependent; for example:

- orthographic sentences, quotations;
- orthographic words;
- abbreviations, names, dates, highlighted words;

etc.

Level 3. Linguistic annotation. This level builds on level 2 by enriching the text with the results of some linguistic analyses; most often in language engineering applications, such analysis is at the sub-paragraph level. For example:

- morphological information;
- syntactic information (e.g., part of speech, parser output);
- alignment of parallel texts;
- prosody markup;

etc.

The TEI standard provides the basis for corpus markup for levels 0, 1, and 2 as well as many elements of level 3.

6.2. Minimum requirements

For each of the levels defined above, the CES will define a minimum set of requirements for conformance to that level, consisting of syntactic requirements and a set of minimal facts required to be marked. Thus a corpus can be identified as conformant at level 0, conformant at level 1, etc.

Three criteria will serve to determine the required minimums:

1. *usability* of the encoding, in terms of providing the information required for the application--here, the CES will be developed following the TEI approach of being minimally prescriptive so as to allow a maximum of flexibility;

2. *reusability* of the encoding, that is, an encoding which makes possible the enhancement of the corpus to higher encoding levels and for other applications; and

3. the *costs of implementation,* such that expensive operations (such as manual identification of a large number of elements) are avoided when possible and the required minimum can be achieved at the lowest possible cost (see section 5.7).

Notes

[1] See, for example, the recent double issue of *Computational Linguistics* on Using Large Corpora, 1993, Vol. 19, 1 & 2.

[2] Other familiar markup languages, such as LaTeX, are at the *markup specification* level (see section 4.2). The user has very little access to the metalanguage level in such languages, since tags and rules governing their use them are pre-defined.

[3] This applies to the TEI interchange standard, as opposed to the TEI local processing standard, which is less restrictive concerning SGML use.

[4] See [Ide94b] for more discussion of this TEI design principle.

[5] EAGLES has, in fact, recently established such a set of proposed POS categories for use in NLP research on western European languages.

[6] These principles are a variant of those proposed for encoding dictionaries when it is desired to be faithful to some printed original (see TEI-P3, "Base Tag Set for Printed Dictionaries"), but it might be possible to extend them to apply more broadly.

[7] There are other types of validation. We can distinguish three kinds of validation:

- *Source validation:* validation against the source, when one exists, to ensure that no unintentional distortion (mistyping, omissions, etc.) has been introduced, as discussed in section 5.3.

- *Structural validation:* validation in the SGML sense, by verifying document structure against the DTD, as discussed in this section.

- *Content validation:* validation against the conceptual document model, to ensure that markup use is appropriate, e.g., that what is marked as a sentence conforms to the user's definition of sentence.

[8] Note that the use of attributes (for example, **<div type=chapter>**) does not solve this problem, since there is no way in SGML to provide a content model for a tag when it appears with a given attribute.

[9] The tags in this example are not necessarily those defined within the TEI. Marking the triple word/pos/lemma is not straightforward within the current TEI scheme.

[10] See the chapter "Structure of the TEI Document Type Declarations" in TEI-P3 for a full description.

References

[Brya88] Bryan, M. *SGML: An Author's Guide,* Addison-Wesley Publishing Company, New York (1988).

[Chil94] Chilsholm, D. and Robey, D. Encoding verse texts. *Computers and the Humanities 28*, 1-3 (1994), (to appear).

[Coom87] Coombs, J.H., Renear, A.H., and DeRose, S.J. Markup systems and the future of scholarly text processing. *Communications of the ACM 30*, 11 (1987), 933-947.

[Gold90] Goldfarb, C.F. *The SGML Handbook,* Clarendon Press, Oxford (1990).

[Herw91] van Herwijnen, E. *Practical SGML,* Kluwer Academic Publishers, Boston (1991).

[Ide94a] Ide, N. and Véronis, J. Encoding Dictionaries. *Computers and the Humanitites 28*, 1-3 (1994), (to appear).

[Ide94b] Ide, N. and Sperberg-McQueen, C.M. The TEI: History and Background. *Computers and the Humanitites 28*, 1-3 (1994), (to appear).

[Pric94] Price, L. Introducing SGML from the TEI perspective. *Computers and the Humanities 28*, 1-3 (1994), (to appear).

[Sper93] Sperberg-McQueen, C.M. and Burnard, L., *Guidelines for Electronic Text Encoding and Interchange*, Text Encoding Initiative, Chicago and Oxford, 1993.