Compiling and annotating corpora in DK-CLARIN
Interpreting and tweaking TEI P5

Jørg Asmussen
Society for Danish Language and Literature
ja@dsl.dk

Jakob Halskov
Danish Language Council
jhalskov@dsn.dk

Abstract

This work-in-progress report discusses the structure and in particular a number of sub-structures of the TEI P5 text header specification which caused certain problems in an ongoing project aiming to gather a new corpus of Danish. The report concludes that certain parts of the TEI P5 need to be both enriched and structured differently in order to become the standard of choice for the DK-CLARIN corpus projects.

The report also presents a general text format which is used as a means of ensuring internal integration of text units within the ongoing multi-institutional project. The format features a primitive segmentation of texts into word and punctuation units. These units have unique xml:ids allowing them to be referenced from layers of annotations, e.g. tokenisation or PoS tagging, which can be added by TEI-enabled tools all operating on the same version of the text proper.

Introduction

Centre for Danish Language Resources and Technology Infrastructure for the Humanities (DK-CLARIN) is a multi-institutional project funded by the Danish Agency for Science, Technology and Innovation[1] (grant number 2136-07-0003). It aims to establish a common infrastructure for language resources and language technology of Danish. It can be seen as a national counterpart to the EU-CLARIN project. However, in contrast to the EU-CLARIN project, which primarily is planning to integrate existing resources on a pan-European scale, DK-CLARIN is not in a preparatory phase. Its objective is – among other things – to compile and annotate a number of corpora. Thus, a synchronic LSP corpus comprising 11 million tokens and a synchronic LGP corpus of some 45 million tokens will be made available online by the end of 2010.

This work-in-progress report focuses on the benefits and challenges of tweaking and interpreting the TEI P5 text header scheme[2] to meet the demands of very heterogeneous texts in the various sub-corpora of the project.

TEI P5 was selected as a joint metadata scheme for all textual resources to achieve internal integration, but also to facilitate future external integration of DK-CLARIN with EU-CLARIN. However, the corpora compiled by the different work packages of DK-CLARIN differ along many dimensions, for example with respect to the time frame (synchronic and diachronic corpora), the language aspect (monolingual and parallel corpora) and the domain specificity (LSP and LGP corpora).

The structure of the header is oriented towards that one used by the BNC (Burnard, 2007) and PAROLE-DK (Keson, 1998a; Keson, 1998b) but tries to avoid idiosyncrasies not covered by TEI P5 as well as modifications of the TEI header schema. However, the common TEI P5 compliant text header needed some interpretation to meet the demands of the various

work packages and their heterogeneous texts (DK-CLARIN also includes corpora of spoken language and multimodal resources, but these are not covered by this report).

Also, a common TEI P5 compliant standard text format needed to be developed. Without such a common format DK-CLARIN language technology tools (e.g. tokenisers, PoS taggers and lemmatisers) would not be able to annotate resources across all the different work packages.

1.0 Corpus-compositional prerequisites

All written text units that are potentially to be included in a future corpus for linguistic purposes are collected in a repository, a Corpus Text Bank, *CTB*. A *text unit* consists of the *text* proper and of some metadata about the text contained in a *header* preceding the text. A text unit is the smallest chunk of text in the CTB and thus is the smallest corpus-compositional unit. The text part of a text unit is either a complete text (usually a shorter one) or a sample taken from a longer text. The CTB is implemented as an XML database, using eXist-db[3] as database management system together with a specially developed web-based viewer, editor, and corpus-composition tool.

The CTB will contain all kinds of written corpus-relevant texts collected as part of the DK-CLARIN project's work package 2, 'Basic written language resources'. Text units from the CTB may be included in one or more specific corpora intended for linguistic research. A *corpus* is a more organised collection of texts compiled on the basis of the text bank for a specific – i.e. linguistic – purpose. Text material being collected for literary purposes or as part of an electronic library or archive may stress other features of the TEI header proposal. Here, the header structure is adapted to the specific needs of *corpus* texts.

2.0 The text header

This section describes the header structure of text units to be collected in the CTB. Text headers (as well as the texts themselves) are structured by means of TEI P5. The following sections describe this structure which is adapted to the needs of integrating various existing corpora or text collections. The collections to be structurally integrated are the *Corpus of the Danish Dictionary* (DDOC, Norling-Christensen and Asmussen (1998)), *PAROLE-DK* (Keson, 1998a) and Keson (1998b)), *Korpus 2000* (Andersen et al. (2002)), other corpus-relevant material gathered at the Society for Danish Language and Literature, DSL, and the Danish Language Council, DSN, as well as the LGP and LSP corpora of written Danish which are compiled as part of the DK-CLARIN project.

The TEI header structure provides extremely flexible means of expressing textual metadata. A wealth of information can be given in a more or less fine-grained way. The following sections describe a header that exactly accommodates the needs of the above-mentioned text collections. In many cases, TEI allows the header to be modified either by augmenting or simplifying it. However, a header with more or less information will still be compatible with the model described here as long as its structure does not conflict with TEI P5 syntax (and semantics) requirements.

Thus, we do not describe a TEI header in general, but the specific header of a potential corpus text in the CTB, expressed by means of TEI.

2.1 Header structure

The header of a text unit provides a structured description of the text contents. Every separate text unit in the CTB has its own header <*teiHeader type="text*">. In addition, a corpus itself has a header <*teiHeader type="corpus*"> containing information which is applicable to

the corpus. The corpus header is not part of this description. To a large extent, a corpus header is a structurally abridged and slightly modified version of a text header that also contains the declaration of value sets for various elements (e.g. a domain taxonomy for LSP texts). The CTB contains value declarations in form of a collection of certain value set files that may be referenced by the CTB header. The remainder of this section describes the components of the *<teiHeader type="text">* element, as used within the CTB.

A TEI header contains a file description (Section 2.1.1), an encoding description (Section 2.1.2), a profile description (Section 2.1.3), and a revision description (Section 2.1.4), represented by the following four elements:

<fileDesc> *(file description) contains a full bibliographic description of an electronic text* as well as the source from which it was derived. ***<encodingDesc>*** *(encoding description) documents the relationship between an electronic text and the source or sources from which it was derived.* ***<profileDesc>*** *(text-profile description) provides a detailed description of non-bibliographic aspects of a text, specifically the languages and sublanguages used, the situation in which it was produced, the participants and their setting.* ***<revisionDesc>*** *(revision description) summarises the revision history for a file* (TEI P5 header specifications[4]).

2.1.1 The file description

The file description *<fileDesc>* contains the following four subdivisions:

<titleStmt> (title statement) groups information about the title of a work as represented in the electronic text sample. ***<extent>*** specifies the size of the electronic text sample in number of words and paragraphs. ***<publicationStmt>*** (publication statement) groups information concerning the publication or distribution of the electronic text sample.
<notesStmt> (notes statement) collects together any notes providing information about a text additional to that recorded in other parts of the bibliographic description.
<sourceDesc> (source description) supplies a description of the source text from which the electronic text sample was derived.

In the following we will focus on the <publicationStmt> and <sourceDesc> elements which we found particularly difficult to use for our purpose, and we will outline the solutions, i.e. interpretations and tweaks, we arrived at.

2.1.1.1 publicationStmt/Availability

The following pattern shows the substructure of the *<availability>* element:

```
<availability status="restricted">
<ab type="academic">
<seg type="availDesc">availDesc</seg>
<seg type="anonymDesc">anonymDesc</seg>
</ab>
<ab type="nonCommercial">
<seg type="availDesc">availDesc</seg>
<seg type="anonymDesc">anonymDesc</seg>
</ab>
<ab type="all">
<seg type="availDesc">availDesc</seg>
<seg type="anonymDesc">anonymDesc</seg>
</ab>
```

</availability>

The text strings in <*ab*> ('anonymous block') elements given under <*availability*> for both restricted (attribute *status* is set to "restricted") and free (attribute *status* is set to "free") give availability information for three fixed user categories: academic users, non-commercial users, and all types of users.

Academic users are defined as users who are affiliated with the DK-CLARIN consortium.
Non-commercial users are academic users not affiliated with the DK-CLARIN consortium, users from educational or governmental institutions.
All users are any type of users including commercial users.

The <*availability*> element requires subordinate <*p*> or <*ab*> elements thus inhibiting more meaningfully structured availability information. The cumbersome solution of using typed <*ab*> and <*seg*> elements thus seem to be the only way of expressing structured availability information, unless TEI P5 is extended.

Two types of values are given in two subordinate <*seg*> elements: The availability description *availDesc* and a description of how to make anonymous private information associated with the text, *anonymDesc*. If availability for any user category is other than "full" or any kind of anonymisation is required, that is if *anonymDesc* is other than "nothing", the availability *status* attribute is set to "restricted", otherwise it is set to "free".

2.1.1.2 sourceDesc

The <*sourceDesc*> element is used to supply bibliographic details for the original source material from which an electronic text sample derives. In the case of DK-CLARIN corpus texts, this may be a book, pamphlet, newspaper, etc. or an electronic source of some (non-TEI) format. Within the <*sourceDesc*> element several sub-structures are available according to TEI. Here, the <*biblStruct*> sub-structure is used in almost the same way as in the PAROLE Corpus (Keson 1998a, Keson 1998b) because it imposes a fixed structure on the bibliographic description and, most importantly, because it allows to distinguish between information concerning the text proper and information concerning the edition (e.g. book, newspaper) from which the text was derived:

<sourceDesc>
<biblStruct>
[...]
</biblStruct>
</sourceDesc>

The <*biblStruct*> element contains the following main elements:

<*analytic*> (analytic level) contains bibliographic elements describing an item (e.g. an article or poem) published within a monograph or journal and – according to the TEI guidelines – not as an independent publication. In the CTB headers, though, it is used for independent publications as well, see below.
<*monogr*> (monographic level) contains bibliographic elements describing an item (e.g. a book or journal) published as an independent item (i.e. as a separate physical object)

According to the TEI guidelines,

> *[in] common library practice a clear distinction is usually made between an individual item within a larger collection and a freestanding book, journal, or collection. Similarly a book in a series is distinguished sharply from the series within which it appears. An article forming part of a collection which itself appears in a series thus has a bibliographic description with three quite distinct levels of information: the analytic level, giving the title, author, etc. of the article; the monographic level, giving the title, editor, etc. of the collection; the series level, giving the title of the series, possibly the names of its editors, etc. and the number of the volume within that series*[5]. (TEI P5 guidelines)

The aim of the bibliographic information for texts which are intended to be included in a corpus, that is the type of texts collected in the CTB, is not to imitate the precision of a librarian but to give an easy way of referring to texts and to probably use bibliographic information in some corpus searches as well. This requires a rather fixed and, to some extent, rigid structure of the bibliographic part of the header, and this is the reason why the <*biblStruct*> structure is used here and not one of the other (less fixed) possibilities of TEI.

The <*biblStruct*> structure can be used to distinguish between the three information levels discussed above in the TEI guideline snippet. Here, only two of the levels are used, namely the analytic and the monographic level. The <*monogr*> element in the <*biblStruct*> structure is obligatory. According to TEI, it seems that in the case of a text being monographic, the <*analytic*> part of the structure should be left out and the text title and author information should be given within the <*monogr*> part of the structure. However, in CTB headers, the <*analytic*> part is considered *obligatory*, no matter whether the text is part of a collection of some kind, i.e. analytic, or a stand-alone publication, i.e. monographic. This is to ensure that all <*biblStruct*> elements in CTB headers have the same structure, so that the text title and author information is always found in the same place, namely in the obligatory <*analytic*> part of the structure.

Within the <*analytic*> structure, <*title*> always gives the title of the text. If the text is part of a collection, e.g. a newspaper article which is part of a newspaper, the *level* attribute of <*title*> is set to "a" which means *analytic*, whereas the <*title*> element in <*monogr*> gives the title of the collection, e.g. the name of a newspaper. If the text is a free-standing book, e.g. a novel, the *level* attribute is set to "m", meaning *monographic*; in such cases the <*title*> element in the <*monogr*> part is left empty.

The author of a text is always given in <*author*> in the <*analytic*> part of <*biblStruct*>. There is one <*author*> element for each author who has contributed to the text. The name of the author is given in a <*name*> element.

If the name has been decomposed into forename and surname, the information is given as *surname, forename(s)*, otherwise the comma is left out. If the name of the author is unknown, the <*name*> element is filled in with an *unknown* symbol, see Section 2.2. A <*name*> element may have a *ref* attribute giving an XML reference to a corresponding <*person*> element in the <*profileDesc*> part of the header where additional info concerning the author(s) can be given.

In the <*monogr*> part, the title of the collection is given if the text is part of a collection, otherwise it is left empty. The name of the editor is given in a <*name*> element (or several elements in the case of multiple editors) which may also have a a ref attribute to a corresponding <*person*> element in the <*profileDesc*>.

2.1.2 The encoding description

The second major component of the TEI header is the encoding description <*encodingDesc*>. This component contains information about the relationship between an

encoded text and its original source. The CTB <*encodingDesc*> element has the following sub-elements:

<*samplingDecl*> (sampling declaration) contains a description of the method used in sampling the text
<*projectDesc*> (project description) describes the aim or purpose for which an electronic file was encoded
<*appInfo*> (application information) records information about the applications which have processed the TEI file.

Of these, we will have a closer look at <*appInfo*>.

2.1.2.1 appInfo

The <*appInfo*> element gives information about all applications or manual procedures by which the text sample has been processed, thereby indicating how and to what extent the text has been enriched with additional or more precise mark-up. The header itself may also be manipulated by such applications or procedures, but this is not registered in the <*appInfo*> element – this may however be recorded under <*revisionDesc*>. The application information helps determining to what extent texts are structurally identical as texts that have been processed by the same bundle of applications and procedures can be considered structurally identical.

```
<appInfo corresp="#textRef">
<application xml:id="appXmlId"
        type="appType "
        subType="appTask"
        ident="appId "
        version="appVersionNumber">
<desc>appDescription</desc>
</application>
</appInfo>
```

The <*application*> element has the following attributes:

xml:id is a unique XML identifier which is referenced by the corresponding annotation in the text.

type specifies whether the task was performed by an automatic application or a manual procedure.

subtype specifies the type of job that was performed on the text by the application or procedure, i.e. whether it has split the text into segments (a segmenter) or annotated segments with further information (an annotator).

ident supplies a unique identifier for the application/procedure.

version supplies a version number for the application/procedure.

The <*application*> element contains an element, <*desc*>, giving a description of the application which is taken from a fixed list of options. The following is an invented example of

what the <*application*> elements could look like in a CTB header:

```
<appInfo corresp="#standard.ctb">
<application type="auto"
        subtype="segmenter"
        ident="dottok"
        version="1.0">
<desc>s-splitter tokenizer</desc>
</application>
<application xml:id="cstLemma"
        type="auto"
        subtype="annotator"
        ident="cestle"
        version="5.3">
<desc>regularizer lemmatizer</desc>
</application>
</appInfo>
```

2.1.3 The profile description

The third component of a TEI header is the profile description <*profileDesc*>. In the CTB, this is used to provide the following elements:

<*creation*> contains information about the creation of a text.
<*langUsage*> (language usage) describes the languages, sublanguages, registers, dialects etc. represented within a text.
<*textDesc*> (text description) provides a description of a text in terms of its situational parameters.
<*textClass*> (text classification) groups information which describes the nature or topic of a text in terms of a standard classification scheme, thesaurus, etc.
<*particDesc*> (participation description) describes the identifiable speakers, voices, or other participants in a linguistic interaction.
In the following, we will focus on parts of the <*textDesc*>, <*textClass*>, and <*particDesc*> elements only.

2.1.3.1 TextDesc

The overall intention of using this part of the TEI proposal is to establish a structure that can contain text descriptions that can be applied to *every* potential corpus text collected in the CTB. Hence, this structure is considered as general and mandatory for every text in the CTB and information from this structure can be used to extract corpora from the CTB. Specialised textual information, which may only apply to *some* texts, is gathered in the <*textClass*> part of the header, see Section 2.1.3.2. Also, the amount of specialised textual information may vary from text to text.

The <*textDesc*> element characterises each text according to a number of situational parameters. In the CTB, the <*textDesc*> structure looks as follows:

```
<textDesc>
<channel mode="tdChannelMode">tdChannel </channel>
<constitution type="tdConstitutionType"/>
<derivation type="tdDerivationType">
        <lang>languageId</lang>
```

</derivation>
<domain type="tdDomainDiscourse">tdDomain </domain>
<factuality type="tdFactualityType"/>
<interaction active="tdInteractActive" passive="tdInteractPassive">
<note type="interactRole">tdInteractRole</note>
<note type="interactAge">tdInteractAge</note>
</interaction>
<preparedness type="tdPrepType "/>
<purpose type="tdPurposeType"/>
</textDesc>

With the exception of the <*interaction*> element no tweaks or interpretation were needed here. The <*interaction*> element contains two subordinate <*note*> elements, one of them indicating the roles of the participants in the communication, that is whether they are experts or laymen; the other <*note*> element gives the ages of addressor and addressee. The former of these pieces of information proved very important when compiling an LSP corpus and the latter could be relevant when compiling an LGP corpus.

However, using a <*note*> element for giving further interaction-related information is not an optimal solution. A straighter way is to use special elements for the needed purposes or to augment the attribute list of the <*interaction*> element. But this would require a modification of the TEI grammar.

2.1.3.2 TextClass

Texts may be described along many dimensions and according to many different taxonomies. No generally accepted consensus as to how such taxonomies should be defined has yet emerged. To accommodate special needs, TEI allows to express more specialised text characteristics by the following elements:

<*catRef*> (category reference) provides either a list of codes or one single code identifying the categories to which the text has been assigned, each code referencing a category element declared in the corpus header or under a separate, persistent URI. In CTB, there is one <*catRef*> element for each dimension, the type of dimension is indicated by the (referencing) value of the attribute *scheme*. CTB does not use lists of codes.

<*classCode*> contains the classification code used for the text in some standard classification system. There is one <*classCode*> element for each classification system.

Using <*catRef*> is the preferred way to give additional textual classifications in all cases where the classification system follows a CTB-internal standard. The pattern to be applied is as follows:

<textClass>
<catRef scheme="myClassification" target="myValue"/>
</textClass>

The <*catRef*> element is repeated for each classification dimension used. In cases where an official classification system is applied, the <*classCode*> element is used instead. The <*catRef*> and <*classCode*> elements should be used as illustrated by the following, invented, example:

```
<textClass>
        <catRef scheme="dk-clarin.eu/ctb/agerel" target="#a-c"/>
        <catRef scheme="dk-clarin.eu/ctb/domain" target="#med"/>
        <catRef scheme="dk-clarin.eu/ctb/genre" target="#ad"/>
</textClass>
```

2.2 Value sets for header standard information

When filling in the header with standard information about the text, some types of information may be undetermined or non-existent, e.g. the name of an author may be simply missing in the header for some reason, that is, it is *undetermined*, or a text may not have a title, that is, its title is *non-existent*. Such incomplete parts of the header could be left out in these cases if permitted by TEI; however, leaving out such parts would obscure whether the information is missing because it is undetermined or because it is non-existent. If the information is undetermined, efforts should be undertaken to occasionally add it, otherwise, if it is non-existent, such efforts would be waste of time. In order to distinguish these two cases, it is recommended to always explicitly state non-existent information by filling in *n/a* (= not applicable) for string and symbol values and *0* (= zero) for integers, in other words never to leave out these parts of a header. However, if the information is undetermined, these parts of a header may be left out which indicates that the missing information occasionally should be added or be marked as non-existent if that is the case.

In the case of undetermined information, it is legal to skip the according part of the header if allowed by TEI; however, for the sake of completeness, it is strongly recommended to state *undetermined* in case of string values and *-1* in the case of integers to indicate that this particular information obviously is missing and should be added if it exists or, if it turns out that the information definitely does not exist, it should be marked as non-existent. To sum up, the following constant symbols are used as values for header elements and attributes, unless otherwise stated further below in this section[6]:

Symbol	Type	Meaning
n/a	String	Info is non-existent
0	Integer	Info is non-existent
Undetermined	String	Info has not been determined yet
-1	Integer	Info has not been determined yet

Table 1: Being explicit about undetermined versus non-existent information.

3.0 The standard text format

The main motivation of defining a *general text format* is to establish a joint basis for all tools that operate on CTB texts. Thus, tools do not need to be configured for a multitude of formats which means that they will be easier and less error-prone to develop and maintain.

Format requirements

1. The format must be expressed by means of TEI P5

2. Annotations should not interfere with the basic format of the text proper

3. The basic format of the text proper should not be biased by interpretations

4. It must be possible to annotate one single text with various (possibly mutually exclusive)

types of annotations, each type appearing as a group of annotations that conceptually belong together

5. Each annotation in an annotation group must be able to refer either to the text proper or to another annotation group which means that layers of annotations, i.e. annotations on annotations, should be feasible

6. It should be possible to store annotations separate from the text proper

7. Several versions of the text proper should be avoided

Consequences

• The text has to be mechanically segmented into basic units
• It must be possible to unequivocally refer to these units
• A generalised, multi-purpose format that needs to be transformed in order to be legible for humans which means that specific viewers and editors must be developed in order to interact with the text

3.1 From source version to base format

Two of the consequences emerging from the requirements were that the text has to be mechanically segmented into *basic textual units* and that it must be possible to unequivocally refer to these units. Mechanical text segmentation is carried out by certain textual surface items, i.e. characters, only. For segmentation purposes characters fall into three categories:

• letters and numbers, i.e. alpha-numeric characters
• whitespace characters
• punctuation characters

Continuous sequences of alpha-numeric characters are considered 'words' even if these segments are not necessarily in accordance with a linguistic definition of a word. Linguistic interpretations are deliberately avoided at this point. 'Words' are
put into <*w*> elements.

Whitespace and punctuation is put into <*c*> elements – character by character – that can be of *type* "s" (space) or "p" (punctuation). A facultative *subtype* attribute may specify some other characteristics of the character in question, e.g. the length of a whitespace. Specifications of the possible inventory of the subtype attribute are not given before it turns out that this attribute is really needed. Standard space characters are not explicitly denoted in the <*c*> elements (i.e. they remain empty) whereas other whitespace characters such as tabs (coded as 	) can be given in the element.

The <*w*> and <*c*> elements are the smallest segments (i.e. *basic units*) of a text. Each of them carries a unique *xml:id* that allows referencing to it from elsewhere. For example, the sentence "De står over for et problem i dag." (*They are facing a problem today*) might look like this after segmentation:

```
<p>
<s>
<w xml:id="y01">De</w>
<c xml:id="y02" type="s"/>
<w xml:id="y03">står</w>
```

```
<c xml:id="y04" type="s"/>
<w xml:id="y05">over</w>
<c xml:id="y06" type="s"/>
<w xml:id="y07">for</w>
<c xml:id="y08" type="s"/>
<w xml:id="y09">et</w>
<c xml:id="y10" type="s"/>
<w xml:id="y11">problem</w>
<c xml:id="y12" type="s"/>
<w xml:id="y13">i</w>
<c xml:id="y14" type="s"/>
<w xml:id="y15">dag</w>
<c xml:id="y16" type="p">.</c>
</s>
</p>
```

This formatted version of the source text is called the text's *base format*. The base format is the standard input format for all tools like tokenisers, sentence splitters, lemmatisers, and taggers of all kinds.

As can be seen, pre-existing mark-up above the <*w*> and <*c*> level is not discarded as long as the source version text complies with the TEI specifications. In this case, the <*p*> and <*s*> tags in the source version are maintained in the base format.

3.2 Layers of annotation

Annotations are given separately from the base format version of the text by a number of <*span*> elements enclosed in <*spanGrp*> elements. The <*span*> elements contain the annotations themselves that are either attached to one single basic textual unit or a number of continuous basic textual units. Attachment is achieved by referencing the *xml:id* units from the obligatory *from* attribute of the <*span*> element and – in case continuous basic textual units are referenced and not only a single one – the facultative *to* attribute. Every <*spanGrp*> contains one type of annotations only. The *ana* attribute of the <*spanGrp*> element refers to the application or method that has produced the annotations, listed in the <*appInfo*> element of the header. Some annotation examples follow.

3.2.1 Tokenisation

Although the segmentation of a text into the base format illustrated in Section 3.1 could be considered as a kind of "blind" or primitive tokenisation, a more linguistic tokenisation often requires a little more sophistication. The following span group structure is an example of how a "proper" tokenisation tool might annotate (and normalize) the base format:

```
<spanGrp ana="#tokenRegular">
<span xml:id="t1" from="#y01">de</span>
<span xml:id="t2" from="#y03">står</span>
<span xml:id="t3" from="#y05" to="#y07">over for</span>
<span xml:id="t4" from="#y09">et</span>
<span xml:id="t5" from="#y11">problem</span>
<span xml:id="t6" from="#y13" to="#y15">i dag</span>
</spanGrp>
```

3.2.2 Part of speech tagging of a tokenised text

Additional layers of annotation can be added by inserting an additional span group containing span elements which refer either directly to segments in the base format or to span elements in other annotation layers. In the following example, a part of speech tagger has processed the same sentence which was tokenised in Section 3.2.1 and has assigned part of speech categories to the token spans (not the segments) in this sentence:

<spanGrp ana="#pos">
PRON
V
PRP
ART
S
ADV
</spanGrp>

Conclusions

TEI P5 is a very powerful specification in many respects, also in the specific respect of annotating corpus texts with relevant metadata. However, as is illustrated in this work-in-progress paper there are certain aspects of the standard which are suboptimal. For example, the way in which structured availability information (see Section 2.1.1.1), bibliographic information (Section 2.1.1.2) and information about the communicative setting (see Section 2.1.3.1) must be represented to comply with TEI P5.

Summing up, TEI P5 is sometimes insufficient in that it does not allow one to express what may seem necessary. In these cases one must make provisional use of semantically empty *<ab>* or *<note>* elements and enrich these with appropriate semantics by adding "type" attributes or other attributes.

Sometimes TEI P5 allows potentially confusing flexibility by using totally dissimilar sub-structures for very similar types of information. A case in point is the way bibliographic information is to be structured according to the standard. Monographies and texts which are part of a collection must conform to two different structures. In order to avoid subsequent search and retrieval issues, the authors decided to reinterpret the intentions of TEI in this case.

Finally, TEI P5 occasionally allows structural flexibility which may become an impediment in the longer run. This is the case with the format for the text proper. Here the standard allows the text structure to be interwoven with so many different annotations that it may become unprocessable. The solution promoted by the authors of this report is to put annotations in *<spanGrp>* elements, i.e. special layers which are independent of the textual basis. This solution presumably also deviates from the intentions of TEI, but it appears to be necessary in order to generalize tools which are to process the texts.

By adopting the modular approach which has made XML such a great success, it should be possible to address these infelicities of TEI P5 while striking a balance between structural flexibility and rigidity.

Notes

1 http://www.fi.dk
2 http://www.tei-c.org/Guidelines/P5/
3 http://www.exist-db.org/
4 http://www.tei-c.org/release/doc/tei-p5-doc/en/html/HD.html
5 http://www.tei-c.org/release/doc/tei-p5-doc/en/html/CO.html.
6 TEI often does not allow "undetermined" as a value. Instead, it may use "unknown" which does not provide the same distinction between the two cases "undetermined" and "n/a".

References

Andersen, M. S., H. Asmussen and J. Asmussen (2002). ”The project of Korpus 2000 Going Public”. In Braasch, A. and C. Povlsen (ed.), *Proceedings of the 10th EURALEX International Congress*, 1, 291–299, Copenhagen. Euralex.

Burnard, L. (2007). *Reference Guide for the British National Corpus (XML Edition)*. Technical report, Research Technologies Service at Oxford University Computing Services. Available at: http://www.natcorp.ox.ac.uk/XMLedition/URG/index.html (accessed 30 September 2009).

Keson, B. K. (1998a). *Vejledning til det danske morfosyntaktisk taggede PAROLE-korpus*. Technical report, DSL. Available at: http://korpus.dsl.dk/e-resurser/paroledoc_dk.pdf (accessed 30 September 2009).

Keson, B. K. (1998b). *Documentation of The Danish Morpho-syntactically Tagged PAROLE Corpus*. Technical report, DSL. Available at: http://korpus.dsl.dk/e-resurser/paroledoc_en.pdf (accessed 30 September 2009).

Norling-Christensen, O. and J. Asmussen (1998). “The Corpus of The Danish Dictionary”. *Lexikos. Afrilex Series*, 8, 223–242.