Guideline: Multiple Hierarchies

Andreas Witt

SFB 441, Eberhard-Karls-Universität Tübingen
ANDREAS.WITT@UNI-TUEBINGEN.DE

May 31, 2007

Introduction

As the title of the Dagstuhl Seminar "Digital Historical Corpora - Architecture, Annotation, and Retrieval" already suggests, corpus architecture and corpus annotation is an important topic for representing (historical) texts. Especially the limitation of SGML-based markup languages to tree structured annotations raises a special problems when dealing with manuscripts: How is it possible to represent overlap. This problem was addressed by the Text Encoding Initiative (TEI) and by several scholars.

This text gives an overview of several techniques for handling the overlap problem. This problem occurs especially when one wishes to digitize and to annotate richly structured historical texts, e.g. manuscripts.

The overlap problem tends to arise in a variety of contexts such as:

- Annotating a narrative: a speech by a character may begin in the middle of a paragraph and continue for several more paragraphs
- Annotating a verse text: the encoder may need to tag both the formal structure of the verse (its stanzas and lines) and its syntactic structure (often nesting within the metrical structure and sometimes crossing metrical boundaries)
- Recording the physical structure of volume, page, column, and line along with the formal or logical structure of chapters and paragraphs or acts and scenes
- Identifying embedded text units (e.g. a play within a play, or a song) that may be interrupted by other matter, if the logical unity of the embedded material ist to be stated explicitly

Any solution proposed so far comes with severe drawbacks in one or more of the following respects:

Dagstuhl Seminar Proceedings 06491
Digital Historical Corpora- Architecture, Annotation, and Retrieval
http://drops.dagstuhl.de/opus/volltexte/2007/1040

- not exhibiting the desired formal simplicity
- lacking capacity to represent all occurring or imaginable kinds of structures
- lacking suitability for formal or computational validation
- rupture with the notations needed for simpler cases

The following is a discussion of the most common methods of handling non-nesting information. It is heavily oriented on the TEI-Chapter which describes ways to deal with non-hierarchical markup and which was partly written by the author of this text.

1 Multiple Encodings of the Same Information

The simplest method of disentangling several conflicting hierarchical views of the same information is to encode it seperately multiple times, each time capturing a single view. This results in several annotation files, one file for each annotation layer. The main advantage of this approach is that each way of interpreting the information is explicitly represented in the data structure and may be processed without complex methods of distangling information. However, for this to work, redundant information has to be encoded for layers where it doesn't play a role. Furthermore, whenever one of the views is altered, one risks inconsistencies in the data if the other views are not updated accordingly. In addition, it is extremely cumbersome to access annotation data from one view while processing data from another view. To circumvent some of these difficulties, it is recommendable to relate the different annotations in an indirect way: If the textual content the different annotations refer to is identical, th etext itself can serve as a linking device for the different annotation layers.

2 Remodeling of the Document Class

Sometimes it is possible to merge several logical hierarchies into one physical hierarchy. Then, there exists a single hierarchy that contains all the information of the separate annotations. However, there is a certain tendency of such structures not to conform to the TEI document grammar, which makes this solution unacceptable for many text-encoding purposes. Furthermore, many overlapping structures exist that cannot be annotated in this way.

3 Boundary Marking with Empty Elements

The idea behind this is that usually a content object is annotated by encoding it as an XML element which encloses a span with its start and end tags. It is, however, also thinkable to mark a span of text using an empty tag at the beginning and another empty tag at the end. These empty tags are called segment boundary elements. There are several variations on this type of encoding:

3.1 Segment-Boundary Elements

In this approach, all segment boundary delimiter elements are of the same generic type. The rest of the information is encoded in attribute values, e.g. a type attribute to indicate the textual feature being encoded, and a position attribute taking the values "start" and "end" to define start and end tag replacements.

3.2 Paired Segment-Boundary Elements

The names of the segment-boundary elements reflect whether they indicate the beginning of a passage or the end of it, whilst the type of feature being annotated is still indicated with an attribute.

3.3 Paired Typed Segment-Boundary Delimiters

The names of the segment-boundary elements reflect both the type of the annotated feature and whether the beginning or end of such a feature is being marked.

3.4 Co-referenced Overlap Terminus (COLT markup)

A special case of typed segment-boundary delimiters in which the type of element used is exactly the same as the type of element that would have been used without the overlap problem, so the tag names can still be TEI-conformant and normal TEI elements can be used, only without content so that they function not as containers but just as an indication of the start and end of the annotated content. Special attributes indicate which is the beginning and which is the end. Additionally, each pair of empty elements that mark the beginning and the end are explicitly linked to each other. Each of the various methods described above (except for COLT) has the disadvantage that it is difficult to tell which start element corresponds to which end element without a complex processing of the text. One way to improve any of the mentioned variations in this respect would be to use the linking attribute "corresp" to associate the boundary element indicating the end of a passage with that indicating its start. This provides all the information needed to reconstruct all the competing hierarchical views in a simple way. In cases of so-called self-overlap that make it unpredictable which start and end tags belong together, the explicit indication of which beginning corresponds with which end has another extrinsic motivation. The usage of the same element names as for the normal container elements has at least two positive consequences:

- human reader of the markup immediately knows what the segment boundary stands for
- the egment boundary element has exactly the same attribute definition as the container element

When using the same element both in its normal role as a container and as a segment boundary delimiter it is advisable to use an attribute other than the general purpose corresp attribute to indicate the corresponding start-or end-tag. Despite their advantages, segment boundary delimiters can impose a burden of cumbersome processing: Since the elements of the analysis are not uniformly represented by nodes in the document tree, they must be reconstructed by highly specialized software in an ad hoc fashion, which is likely to be difficult and may be error prone. The method disguises the logical relationship between the beginning and the ending of each logical element, making it impossible for standard validation software to provide validation in the same way as elsewhere. Using grammar based schema languages, it is not possible to define a content model for the range limited by empty elements. Rule-based schema languages (e.g. Schematron) can be used instead to define further constraints, they permit a sequence of certain elements between empty elements to be legitimized or prohibited.

4 Fragmentation of Elements and Reconstitution of Virtual Elements

This approach breaks up what might be considered a single element into multiple smaller elements, in order to make it fit within the hierarchy. If one wants to count the occurrences of a content type by counting the corresponding elements, this is a drawback, but a rather minor one if element being broken up is used primarily to signal some characteristic rather than some countable object, as is often the case in linguistics. However, since the technique of fragmentation is a general technique to avoid overlapping markup, the effect of introducing artificially new instances of a particular element must be kept in mind. The advantages of this approach lie in its simplicity and the fact that at least one of the competing hierarchies can be processed in the standard manner. The technique of fragmentation is often complemented by the technique of virtual joins that may be used to combine objects in the text to a new hierarchy. The major advantage of this method that it allows all the hierarchies in the text to be handled explicitly. The major disadvantages include being forced to privilege one hierarchy over the others and to do special processing to reconstitute the elements of the other hierarchies.

5 Stand-off Markup

The classic use of markup is characterized by embedding the annotation in the text, whilst this alternative approach separates the text and the annotation, leading to a structure called stand-off annotation. Stand-off markup for each annotation level can be put into a new document tree whose nodes are XML elements which do not contain textual content, but rather links to another layer (i.e. a node in another XML document or a span of text). In some respects, this

can be seen as a generalization of the virtual joins where not only contents of elements are joined, but also ranges between points within the document. This approach can be subdivided according to different criteria:

5.1 The Link Base

If the link target is a document that already contains markup, the new hierarchy can be built up by reference to this base annotation, e.g. at the end of the same XML-file. This makes heavy use of XInclude in a way similar to the use of join elements. The use of XInclude is recommended because with its help it is possible to specify attributes on the created elements and there is off-the-shelf software that will perform XInclude processing without much ado. If the link base contains only plain text, the range of text to be annotated is indicated by character offsets.

5.2 Number of Link Target Files

Often, a single dedicated annotation layer is used as the link target of all the other annotations. However, it is also possible to freely interlink several layers using this technique. Stand-off Markup has several advantages over embedded annotations:

- different annotation files can contain potentially infinitely many different layers of information
- on a fixed source text, independent parallel coders can produce independent annotations As the title of the Dagstuhl Seminar "Digital Historical Corpora - Architecture, Annotation, and Retrieval" already suggests, corpus architecture and corpus annotation is an important topic for representing (historical) texts. Especially the limitation of SGML-based markup languages to tree structured annotations raises a special problems when dealing with manuscripts: How is it possible to represent overlap. This problem was addressed by the Text Encoding Initiative (TEI) and by several scholars.

  This text gives an overview of several techniques for handling the overlap problem. This problem occurs especially when one wishes to digitize and to annotate richly structured historical texts, e.g. manuscripts.

  The overlap problem tends to arise in a variety of contexts such as:

- annotation files can be distributed without distributing the source text
- discontinuous segments of text can be combined in a single annotation
- possible to produce annotations of a text even when the source document is read-only

However, again there are several drawbacks:

- new stand-off annotated layers require a separate interpretation
- layers — although separate — depend on each other (at least on the base or source layer)
- information may be difficult to access using generic and stable methods
- standard document grammars can only be used for a base level containing both markup and textual data
- standard parsing or editing software cannot be employed, special software for this purpose not readily available

6 Non-XML-based Solutions

Many non-XML methods of encoding a text either solve or do not suffer the problem of the inability to encode overlapping hierarchies:

6.1 SGML feature CONCUR

CONCUR is an optional feature of SGML for annotating multiple hierarchies. However, it has never been implemented properly and is not fully compliant with the intentions behind SGML.

6.2 MECS or TeXMECS language

The MECS (Multi Element Code System) allows overlapping ranges within documents. However, since the tag sets used or needed for a certain annotation task are sometimes quite heterogeneous, this is only part of the solution.

6.3 LMNL meta-language

In addition to overlapping ranges, LMNL (Layered Markup and Annotation Language) also allows for connecting the element names to corresponding annotation levels, also resolving the problem od heterogeneous tag sets. It provides an elegant and somewhat ideal solution for the problems discussed here, but hasn't been implemented yet and will be hard to implement because of its power that reaches beyond even that of SGML. Since the TEI standard is currently based on XML, and these approaches are not, none of these solutions even though potentially promising, can be recommended for humanities computing at the present time.

Additional information

The problem of overlapping markup has been described by different authors. This description was oriented on the TEI-chapter on "Multiple Hierarchies". Another good but relatively old resource for this topic is "http://xml.coverpages.org/hierarchies.html".

At the current time the proceedings of the annual Extreme Markup Languages conference (http://www.extrememarkup.com) is the most important source for getting up to date information on this problem. In August 2007 this conference devotes a one day workshop to "".

Additional information

The problem of overlapping markup has been described by different authors. This description was oriented on the TEI-chapter on "Multiple Hierarchies".

Another good but relatively old resource can be found on the WWW under the address: "http://xml.coverpages.org/hierarchies.html". At the current time the proceedings of the annual Extreme Markup Languages conference (http://www.extrememarkup.com) is the most important source for getting up to date information on this problem. In August 2007 this conference devotes a full-day for the "International Workshop on Markup of Overlapping Structures" (http://www.extrememarkup.com/extreme/overlap/index.html).