Andreas Witt

Meaning and interpretation of concurrent markup

Introduction

The difficulty of annotating not hierarchically structured text with SGML-based mark-up languages is a problem that has often been addressed. Renear et al. (1996) discuss one of the basic assumptions about the structure of text data: the "OHCO-thesis", which states that text consists of an ordered hierarchy of content objects, and show that this assumption cannot be upheld consistently: A number of texts contradict this OHCO-thesis.

The options to represent data which correspond to different structural hierarchies are discussed in detail by the TEI (see Barnard et al. 1995). One type of annotation mentioned by the TEI-guidelines consists of separately annotating the original data according to different document grammars. "The advantages of this method of markup are that each way of looking at the information is explicitly represented in the data, and may be processed in straightforward ways, without requiring complex methods of disentangling information relevant to one view from information relevant only to other views." (Sperberg-McQueen & Burnard 1994, p.775f.) The problem, however, is that separate annotations do not allow for establishing relations between the annotation tiers. It will be shown that a model for the meaning and interpretation of documents, marked-up once, can be extended to documents concurrently marked-up with several annotations.[1]

Model of representation

The model of representing complex structured text is itself based on the data to be structured. This data can be annotated according to specific document grammars. For a simple case there is one document grammar and one annotation. This means all SGML documents and all valid XML documents are represented by this model. In addition to this standard representation every document can be structured according to further document grammars. The maximal number of annotations is not restricted, so that the model permits a steady expansion of the annotations which are used. Therefore, the model can be viewed as an open model. As a result the annotated text data fuses different levels of annotation together.

As a prerequisite the preparation of the data must be done in a way that a separate annotation of relevant phenomena is possible. The data which will be annotated is categorized as having a status of primary data. The primary data consists of the yet to be annotated text. Accordingly, the mark-up will be categorized as secondary data or meta-data.[2] The primary data is used in two ways: On the one hand, the primary data is the subject of several (possible) annotations, on the other hand it forms a link between the levels of annotation. It is important to note that the second way of using the primary data allows for linking the independent annotations without introducing explicit links.

After the stipulation of the primary data individual document grammars can be developed and applied which each pertain to a single level of annotation. For the annotation of linguistic data this could be e.g. the level of morphology, the level of syllable structures, a level of syntactic categories (e.g. noun, verb), a level of syntactic functions (e.g. subject, object), or a level of semantic roles (e.g. agent, instrument).

As already mentioned this annotation technique data allows for the introduction of an unlimited number of concurrent annotations, but there exists a constraint: when designing the document grammar it is necessary to consider that the primary data is the link between all layers of annotation. This has a direct consequence for the modelling process: Even if parts of the primary data are irrelevant for one of the tiers of description, the data must exist as primary data in the annotation. This contradiction can be solved by introducing a special mark for this irrelevant primary data in the document grammar according to the phenomena. This mark allows one to represent the corresponding passages as primary data in a technical way. Such a mark can be done for example by using the element `<ignore>`. While interpreting the distinguished data these marked sections will be filtered out. Durand et al. (1996) already discussed a similar solution where a classification on a more abstract level of representation should happen.

The compilation of document grammars which are used for different annotations, can be seen as a pool of individual units or as a collectively structured source of knowledge. Not only the structure of document grammars, but also the schema language used (e.g. DTD, XSchema) are irrelevant to the process of annotations.

Knowledge representation of annotated text

In general, annotated text consists of content and annotations. Annotations are used on a syntactical level. Therefore they are used for assigning a meaning to (parts of) a document. While developing a document grammar the focus should be centred on the content. This point of view is expressed by Sperberg-McQueen, Huitfeldt and Renear (2000). They show how knowledge which is syntactically coded into text by annotations can be extracted by knowledge inference. After summarizing this approach, it will be shown, how this technique can be expanded so that it can be used for inferences of separately annotated and implicitly linked documents - documents marked-up according to different document grammars.

Documents

Sperberg-McQueen et al. (2000) regard annotated documents as a compilation of knowledge which can be represented and can be used for inference. Illustrating their approach they use the programming language "Prolog" to represent this knowledge. A XML-Document without cross relations can be represented as a tree. A representation of annotated text in Prolog looks like the following:

```
node([x, y, z], element(gi)).

attr([x, y, z],attr-name, 'att-value').
```

A predicate with two arguments, called `node` (or short notation in Prolog `node/2`), is used. The first argument is a list of digits to identify each node of the tree representation of documents. The second argument is more complex. It consists of the functor `element` or `pcdata` and an argument for the name of the element or the textual content. Furthermore, a predicate of three arguments called `attr/3` is used for representing attributes. Attributes are also related to nodes (argument 1). They have a name (argument 2) and a value (argument 3). Such a representation of a document is a very good basis for automatic inference of relations which exist between the nodes. If e.g. `infer/2` is called with an address, all active features can be printed out:

```
      infer(Property,[x, y, z]).

      Property = a;
```

If `infer/2` is called with a concrete feature and a variable at the place of the address, all locations are printed out which apply to these features.

Other frameworks for knowledge representation of annotated documents also exist. Welty and Ide (1999) introduced an approach, where the knowledge of documents is pasted into a knowledge representation system. However, so far it has not been shown how to use differently annotated documents as a base for inferences.

Relations between separate annotations

The described model of knowledge representation can only be used for single documents. However, it will be shown, that this model can easily be expanded, so that it is applicable for the inference of relations between several separately annotated XML-documents with the same primary data.

In order to use the model for several different annotations the representation of an absolute system of references must be introduced. This absolute reference system is already given in the data: the parts of text which are saved as "PCDATA", i.e. the primary data as defined above, is ideally suited for this task. This means that the string which is identical in all documents serves as the absolute system of references.

The primary data forms a string of a fixed length. The representation of the annotation tiers is an absolute basis of relations. This basis must renounce the predicates `node/2` and `attr/3` in favour of the predicates `node/5` and `attr/6`. The three additional arguments are used for the reference on the level of annotation (comparable to the concept of namespaces in XML), for marking the start and the end of the annotated textual content.

The extension of the original model in this way allows for inferences of relations between different concurrent annotations, e.g. regarding the separate annotation of morphemes and syllables might show that these units are not compatible.[3] Since the arguments originally introduced are reused, all the inferences of the original model are still possible.

Advantages and Perspectives

The outlined architecture has many advantages. The model allows for structuring text according to multiple concurrent document grammars without workarounds. Furthermore additional annotations can be subsequently included, without changing already established annotations. The annotations are on the one hand independent of each other, on the other hand they are interrelated via the text, allowing for the inference of relations between different levels of annotation. The final advantage to be mentioned is that the compatibility of several or all annotations used can be proven automatically. This can be done using a technique originally developed within linguistics, namely unification.

Endnotes

[1] Barnard, David et al. (1995). Hierarchical Encoding of Text: Technical Problems and SGML Solutions. In: Ide and Véronis (1995), 211-231

[2] Durand, David, Elli Mylonas and Steven DeRose. (1996) What Should Markup Really Be?. *ALLC/ACH* 1996.

[3] Ide, Nancy and Jean Veronis (Eds., 1995). *TEI: Background and Context*. Kluwer.

[4] Renear, Allen, Elli Mylonas and David Durand (1996). Refining Our Notion of What Text Really Is: The Problem of Overlapping Hierarchies. In: *ALLC/ACH* 1992. Clarendon.

[5] Sperberg-McQueen, C. M. and L. Burnard (Eds., 1994) *TEI-Guidelines (P3)*.

[6] Sperberg-McQueen, C. M., Claus Huitfeldt and Allen Renear (2000). Meaning and interpretation of Markup. In: *Markup Languages* 2.3, 215-234. MIT Press.

[7] Welty, Chris and Nancy Ide (1999). Using the right tools: enhancing retrieval from marked-up documents. In: *Computers and the Humanities*. 33(10). 1999. Kluwer. 59-84.

[8] Witt, Andreas (2002) *Multiple Informationsstrukturierung mit Auszeichnungssprachen. XML-basierte Methoden und deren Nutzen für die Sprachtechnologie*. Ph.D. thesis, Bielefeld University.

Notes

[1] The work presented here is described in more detail in Witt (2002).

[2] The terminus "meta-data" is preferred by the author, because "secondary data" presupposes a characterization as less important data. However, "meta-data" is far from being the ideal term: it causes an ambiguity, since it is also used for the information typically contained in the headers of annotated documents.

[3] Other relations of concurrent markup can be, for example, compatibility, identity, and inclusion.