Multiple Markups in XML Documents

Luis Arévalo, Antonio Polo, Miryam Salas, and Juan Carlos Manzano

Department of Computer Science. University of Extremadura[1]
{ ljarevalo, polo, juancman,miryam }@unex.es

Abstract. The nesting rule for markup tags in XML brings about, on occasions, that different interpretations of the same text may not be represented in the same document. This situation leads to a duplication of the text into different files, each one with the appropriate markup to that interpretation. In this work, a technique, which we will regard as *meta-markup*, is put forward in order to solve this problem. The technique consists in storing the information of each mark -either beginning tags or end tags- in a single element. The mechanisms to integrate all the markups in a single file are shown, and also how to get -from it- each independent markup through XSL transformations. This solution allows an integration of multiple knowledge of the document that may be applied in problems either as versioning or as evolution.

1 Introduction

The standard XML[1] has consolidated as a tool for information exchange and representation due to its flexibility. The specification for XML documents needs a correct nesting for tags. On occasions, such nesting might not be fulfilled, for example, when different interpretations of the same text are to be represented within the same document. If each interpretation is reflected through a specific markup, the combination of different markups in the same document may bring about incorrect tag nesting.

This work starts by showing an example illustrating the problem and the current solutions. Then, we introduce a technique that allows the transformation of a XML document into an equivalent one, called meta-markup. Fourthly, it is shown that, through this technique, multiple markups for a text may be carried out by complying with the XML specification. Finally, the conclusions and future works are introduced.

2 Motivation

Let's imagine that we want to carry out a film adaptation of Shakespeare's play *Hamlet*. We have an adaptation of the dialogues of the play in XML format[2]. This dialogue must be completed with new information (for example with text and the camera shots) in the document so that its contents not be affected. The problem we face is synchronising the text with the shots to be shown at each moment. This problem

[1] This work has been financed by the Spanish CICYT project “TIC2002-04586-C04-02”

J.M. Cueva Lovelle et al. (Eds.): ICWE 2003, LNCS 2722, pp. 222–225, 2003.

might be solved through the labelling of the original document with shot markups. However, the labeling in fig.1 results in a XML document, which is not well formed, because of its incorrect tag nesting. A possible solution for this problem might be to duplicate the text into different files, each one with the appropriate markup for each interpretation but this solution does not allow a single representation integrating all the different markups in the same document.

```
.....
<SPEECH>
    <SHOT id="shot10-g1" description="general overview">
    <SPEAKER>RODERIGO</SPEAKER>
    <LINE>Tush! never tell me; I take it much unkindly</LINE>
    <LINE>That thou, Iago, who hast had my purse</LINE>
    <LINE>As if the strings were thine, shouldst know of this.</LINE>
</SPEECH>
<SPEECH>
    <SPEAKER>IAGO</SPEAKER>
    <LINE>'Sblood, but you will not hear me:</LINE>
    </SHOT>
    <SHOT id="shot24" description="Iago movement">
    ......
</SPEECH>
.....
```

Fig. 1. XML document with multiple markups.

The need to store different markups for a text within the same document has been studied by the Metalanguage Working Group of the Text Encoding Initiative (TEI) [3]. Different research works have been also carried out in [4, 5, 6, 7]. Many of these solutions have been developed mainly for research activity in linguistic fields, where it is necessary to define different interpretations based on the physical and logical characteristics of the text and their relationship. Our proposal is a generic solution based on empty elements [3].

3 Meta-markup Technique

Our solution allows us to transform any XML document into another equivalent one, through the substitution of each opening or ending tag by an element standing for it, that we will refer to as *metamark*. The meta-markup function, M (1), is defined as:

$$\begin{gathered} \mathrm{M}: \chi \rightarrow \chi' = \mathrm{M}(\chi) \subseteq \chi \\ \forall D \in \chi, D = E + T \rightarrow \mathrm{M}(D) = D' = E' + T \end{gathered} \tag{1}$$

Where:

- χ is the type of every XML document well formed, and χ' is the type of every metamarked XML document.
- D is any well formed XML document that we will express as D=E+T, being T an initial text to which a set of E tags are added, in conformity with the XML specification and that represents an interpretation of the initial text T.

- D' is the metamarked D document, in which each E tag has been substituted for its corresponding metamark, so that D'=E'+T, being E' the set of metamarks obtained from this transformation process.

Every metamarked document D' follows the definition of an XML-Schema that we will call *basic schema.* In order to obtain the D' document, each opening or closing tag is substituted for the element MM:oe (MM:ce). For each element MM:oe or MM:ce it is only necessary to store the name of the mark (attribute nm) completing the information of an opening or closing tag. These metamarks are defined as empty elements in the document, except when an opening tag has attributes; in this case the element "MM:oe" may include several elements "MM:at", each one including the information of one attribute used in the original opening mark, with its name (nm attribute) and corresponding value (attribute value). The location of each metamark in the document must be the same as the one of the tag from the original document for which it stands. The original document can be obtained from the metamarked document through the definition of an inverse function M^{-1}(2).

$$M^{-1}: \chi' \rightarrow \chi$$
$$\forall D' \in \chi' \rightarrow M^{-1}(D') = E' + T = D \tag{2}$$

4 Definition of Multiple Markups or Dimensions in a Document

The previous technique can be used to combine, in a single file, different interpretations D on the same text T, represented by different markups E in the form D=T+E. In order to do so, we apply the meta-markup function to each interpretation and combine all the metamarks in a single document, obtaining a document D' represented as D'=E'+T. N interpretations of T would be (3):

$$T\begin{Bmatrix} D_1 = T + E_1 \\ D_2 = T + E_2 \\ \\ D_n = T + E_n \end{Bmatrix} \xrightarrow{M} \begin{Bmatrix} T + E'_1 = D'_1 \\ T + E'_2 = D'_2 \\ \\ T + E'_n = D'_n \end{Bmatrix} \quad T + \cup_{1 \le i \le n} E'_i = T + E' = D' \xrightarrow{M^{-1}/\dim_i} T + E_i = D_i \tag{3}$$

Through the *join* function (∪), we obtain a meta-markup document D' with the metamarks E'_i resulting from each interpretation D_i. Each D_i, will be distinguished by an identifier we will call *dimension*. Accordingly, in order to store the information from the different markups it is necessary to modify the basic schema. A new attribute, "dim", has been added to the three meta-tags we put forward before (oe, ce, at). This attribute will represent for which dimension or dimensions a tag is valid. Besides, we will add a new metamark called "mm:dim" to store the information from each dimension defined over the text.

The validation of a metamarked document will consist, on the one hand, in checking whether it is valid regarding the basic schema, and, on the other hand, in checking whether it is well formed and valid for each dimension regarding the original schema.

An example of transformation of an XML into a metamarked document with two dimensions is shown in fig. 2.

5 Conclusions and Future Works

In this work it has been put forward a technique that might be applied in those situations in which one may wish to store multiple interpretations within a single document or to solve tag nesting problems. The advantages of meta-markup technique are: *1)* it is only necessary to define an XSL transformation sheet to retrieve the different markups in the document and *2)* all the XML documents in the system can be easily turned into metamarked documents using an XSL transformation sheet. The next research steps will be to measure the performance of this technique; to study queries based on the relationships among two or more interpretations and to apply this technique to the development of a system of XML documents versioning.

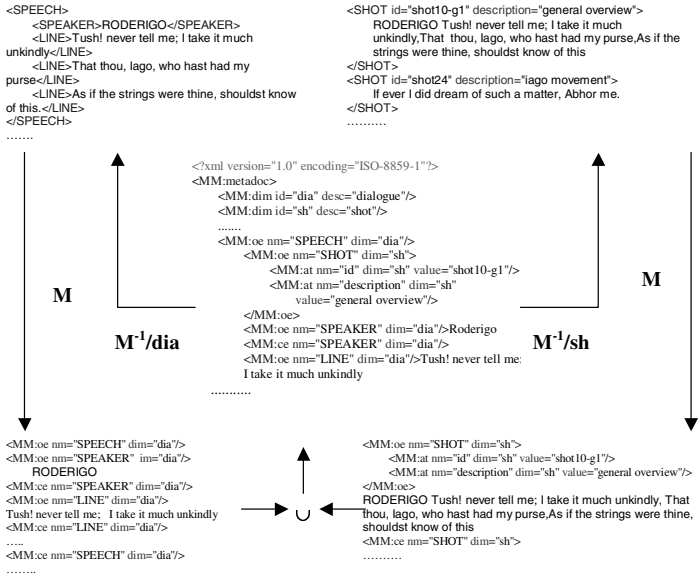

Fig. 2. Validation and retrieval of a metamarked document.

References

1. XML. Available in http://www.w3.org
2. Shakespeare's Plays in XML Format. v2.00. J.Bosak.
3. TEI. Multiple Hierarchies. http://www.tei-c.org/P4X/NH.html
4. Markup Languages and (Non-) Hierarchies. Technology Reports. Cover Pages (Oasis). http://xml.coverpages.org/hierarchies.html. September 2002.
5. TexMECS. An experimental markup meta-language for complex documents. Claus Huitfeldt, Sperberg-McQueen, M. C. 2001
6. Concurrent markup for XML documents. Patrick Durusau, Matthew Brook O´Donnell. XML Europe 2002.
7. The Layered Markup and Annotation Language (LMNL). Tennison, Piez, Wendell. Extreme Markup Language 2002.