

# Preserving Information About Linearization in Document Graphs

**Lars G. Johnsen** (*lars.johnsen@lili.uib.no*)

University of Bergen

**Claus Huitfeldt** (*Claus.Huitfeldt@fil.uib.no*)

University of Bergen

Many operations are more conveniently performed on a graph representation than on a linear representation of a marked up document, and vice versa. Therefore, it is sometimes important to ensure that no relevant aspect of the information contained in a document represented in one of these forms is lost or distorted when the document is converted to the other form.

Conventional methods for converting between XML documents and their graph representations [W3C 2000] are typically seen to preserve such information; standards and methods have been established for ensuring what is in most contexts considered full preservation of all relevant aspects of the linearization [W3C 2001].

However, what is considered relevant may of course vary, depending on context of use. It would probably be hard to find serious arguments to the effect that literally all aspects of the linear representation of a document are relevant for any generally interesting use. Typically, conventional conversion methods are not guaranteed to preserve e.g. attribute order, declaration order, and insignificant whitespace. But it is not hard to find complaints about, for example, lack of preservation of attribute order in certain applications.

Our focus in this paper is on methods for the preservation of element serialization order in marked up documents which make use of mechanisms for representing non-hierarchical complex structures such as overlapping, discontinuous and virtual elements. (For convenience, we use the term "complex structures" to refer to such phenomena.) We do not wish to claim that preservation of element order is always or even generally relevant, our aim is limited to providing a method for such preservation in cases where it is considered relevant.

The customary graph representation of XML is in the form of an "XML tree", a restricted kind of directed acyclic graph (DAG). More specifically, XML trees are DAGs with single parenthood and total ordering on leaf nodes. For certain

purposes, however, a different kind of graph representation has been proposed, the so-called Goddag [Sperberg-McQueen and Huitfeldt 2000]: Roughly, Goddags are like XML trees except that they allow multiple parenthood and do not require a total ordering on leaf nodes; leaf nodes may be ordered only relative to their immediate parents. (Thus, XML trees constitute a subset of Goddags.)

For certain purposes this data structure provides a more convenient representation of complex structures than XML trees. Documents using different XML mechanisms for representing such structures in linear form (e.g. milestones, fragmentation, virtual elements etc. [Barnard et.al. 1995, Sperberg-McQueen and Huitfeldt 1999]) can be mapped on to Goddags, though not without knowledge of application-specific semantics of the markup vocabulary. The experimental markup system TexMecs [Sperberg-McQueen and Huitfeldt 2001] offers mechanisms for the representation of complex structures which can be mapped on to Goddags independently of such knowledge.

However, in both cases, i.e. whether the graph is built from XML or TexMecs, reserialization from the graph is not in general guaranteed possible without changes to the structure and order of elements in the original linearization. For example, if an XML document has used milestones or fragmentation of elements to represent overlapping elements it is possible to build a Goddag representing the non-hierarchical structure of the document. But when reserializing back to XML, the Goddag does not contain any information about which elements to represent as milestones or as fragmented elements.

Similarly with TexMecs: Some element structures can be represented by alternative serialization constructs, and the Goddag as currently defined does not preserve information about the choice of construct in each particular case. In TexMecs the problem is made more severe by the fact that the graph does not, in the case of e.g. virtual or discontinuous elements, preserve complete information about the serial order of elements in the original input.

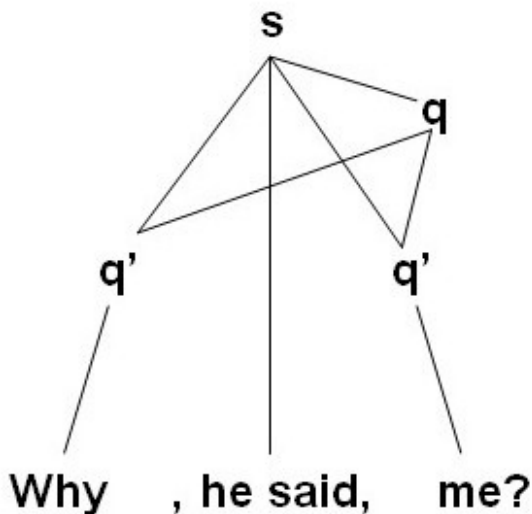
Consider the following example, marked up according to TexMECS, and illustrating how a discontinuous constituent element `<q>` may be represented.

```
(1) <s|<q|why|-q>, he said,<+q| me?|q>|s>
```

In TEI-based XML, the example could e.g. be marked up as:

```
(1') <s><q part="I">Why<q/>, he said,<q part="F"> me?</q></s>
```

The resulting Goddag, whether based on the XML or the TexMecs input, would normally look like this:<sup>1</sup>



Since the second leaf node (containing the string ", he said,") does not share any parent with either of the two other leaf nodes, it is not ordered with respect to these. Therefore, the linearization in (1) is equivalent to the following two linearizations, both placing the second leaf node in a position relative to the other two that it does not have in (1):

- (3) <s|<q|Why me?|q>, he said,|s>
- (4) <s|, he said,<q|Why me?|q>|s>

Thus, the Goddag in (2) would be the same whether built from (1), (3) or (4). Similarly, a choice whether to linearize (2) in the form of (1), (3) or (4) will either have to be arbitrary, or based on purely practical considerations.

A solution to the linearization problem lies, we propose, in considering the Goddag used for representing marked up text as a path ordered directed acyclic graph; a Podagra. Building a Podagra from (1), we get a graph consisting of three paths, in the order indicated as follows:<sup>2</sup>

- (5)
  - 1. s → q → "Why"
  - 2. s → ", he said,"
  - 3. s → q → " me?"

Building a Podagra from (3), however, produces the following path order:

- (6)
  - 1. s → q → "Why me?"
  - 2. s → ", he said,"

whereas from (4) we get the following paths:

- (7)

- 1. s → ", he said,"
- 2. s → q → "Why me?"

The Podagras (5), (6) and (7) all correspond to the Goddag (2), but each maps uniquely to the linearizations (1), (3) and (4), respectively.

In the full paper, we will present an algorithm yielding Podagras from TexMECS documents containing different linearizations also of overlapping and virtual elements. We intend to show how path ordered Goddags can faithfully restore the original linearization of such documents.

Since TexMECS is a purely experimental markup language, these results may be said to have limited practical relevance. However, a number of projects currently build Goddags from XML-encoded documents (by the use of application-specific semantics). Therefore, we also hope to indicate how the proposed method may be used for preservation of the original linearization of XML documents using well-known methods for representation of overlapping, discontinuous and virtual elements.

- 
- 1. For simplification, we are consciously ignoring certain unresolved issues concerning the representation of discontinuous elements in Goddags [Huitfeldt and Sperberg-McQueen 2006].
  - 2. The simplicity of the example allows us to indicate nodes by their generic identifiers. I.e. the three occurrences of "s" all indicate the single node labelled s, and the two occurrences of "q" indicate the single node labelled "q".

## Bibliography

Barnard, David, Lou Burnard, Jean-Pierre Gaspart, , C. Michael Sperberg-McQueen, and Giovanni Battista Varile. "Hierarchical Encoding of Text: Technical Problems and SGML Solutions." *The Text Encoding Initiative: Background and Contents*. Ed. Nancy Ide and Jean Véronis. 1995. 211-231.

Huitfeldt, Claus, and C. M. Sperberg-McQueen. "Representation and Processing of Goddag Structures: Implementation Strategies and Progress Report." *Proceedings of Extreme Markup Languages 2006*. 2006. <<http://www.idealliance.org/papers/extreme/proceedings/>>

Sperberg-McQueen, C. M., and Claus Huitfeldt. "Concurrent Document Hierarchies in MECS and SGML." *Literary & Linguistic Computing* 14.1 (1999): 29-42.

Sperberg-McQueen, C. M., and Claus Huitfeldt. "TexMECS: An Experimental Markup Meta-language for Complex Documents." 2001. <<http://decentius.aksis.uib.no/mlcd/2003/Papers/texmecs.html>>

Sperberg-McQueen, C. M., and Claus Huitfeldt. "GODDAG: A Data Structure for Overlapping Hierarchies." *DDEP-PODDP 2000*. Ed. P. King and E. V. Munson. Lecture Notes in Computer Science 2023. Berlin: Springer, 2004. 139-160.

W3C. *Document Object Model (DOM) Level 1 Specification*. The World Wide Web Consortium, 2000. <<http://www.w3.org/TR/REC-DOM-Level-1>>W3C Recommendation, September 2000

W3C. Ed. J. Boyer. *Canonical XML*. 2001. W3C Recommendation, March 2001