

Concurrent Markup Hierarchies: a Computer Science Approach

Ionut Emil Iacob (eiaco0@csr.uky.edu)

University of Kentucky

Alex Dekhtyar (dekhtyar@cs.uky.edu)

University of Kentucky

Abstract

It is known that text has not, in general, a regular structure. However, since its invention and despite the fact that it represents hierarchical structures, XML has gained a lot of popularity among humanities researchers: XML is easy to use and it comes with a handful of free processing tools. A variety of solutions were proposed to represent overlapping structures in XML. More or less easy to maintain from the point of view of data management, none of these solutions provides full support for two of the most demanded processing tasks: querying and presentation (XSL-like transformation).

We propose a processing framework for complex document-centric XML which generalizes the traditional way of XML data management to support overlapping markup processing. Our framework provides support for overlapping structures representation in XML, querying, authoring, and presentation of overlapping hierarchies.

1. Introduction

The newborn TEI Overlapping Markup Special Interest Group comes to support the fact that overlapping XML structures are of great interest for the text encoding community. Why is XML so popular? First at all, XML is the legitimate descendant of SGML, which was also popular among humanists. Then there is the fact that XML comes with a handful of processing (free) tools. This fact is clearly expressed by TEI's "Strategic Considerations in Migration of TEI Documents from SGML to XML". More specifically, DOM, SAX, XPath, and XSL and the companion software are very attractive for humanities computing. In addition, XML is flexible, intuitive, and readable: it is text, isn't it? However, there is an annoying detail about XML that does not fit into the same picture with text encodings: XML allows only properly nested markup structures. However, overlapping structures (concurrent

hierarchies) often occur in applications. Czymiel points out that the proposed solutions for the overlapping markup problem fall in three categories: XML based workaround (milestones and fragmentation suggested by TEI in Sperberg-McQueen & Burnard), new markup languages such as LMNL (Tennison et al.), MECS (Huitfeldt), and TexMecs (Huitfeldt & Sperberg-McQueen), or content and structure separation (standoff markup, JITTs (Durusau & O'Donnell)). None of the solutions previously presented contains complete answers for the problems of management of concurrent XML data.

Part of the problem is the correct identification of what is the "problem of overlapping markup". For some researchers in humanities, the problem lies in determining the markup elements that can overlap each other, and in devising a way to apply one of the TEI-based suggestions, such as milestone elements. For others, the problem lies in specifying the correct order in which elements overlap, e.g., determining whether paragraph markup must commence inside or outside the page markup.

Such considerations are of importance because they are dictated by the nature of the documents and encoding under consideration. Yet, successful resolution of such issues for specific projects does not constitute solving the overarching problem of representation, storage, management and querying of overlapping markup. Approaching the general solution of such a problem requires a change of a viewpoint from that of a humanities scholar working on document encoding, to the viewpoint of a computer scientist striving to provide generally applicable methodology, algorithms and software tools.

In this paper, we address the issues related to overlapping markup in document-centric XML documents from the perspective of computer scientists. We show that, in general, the framework for processing multihierarchical markup is the same as the framework for processing single-hierarchy XML documents. The difference comes not in the laundry list of tasks, and not in overall organization of the framework, but rather in the approaches to solving individual subproblems/tasks within it.

The processing framework we describe below covers all the core XML processing tasks: representation and parsing, data structure, querying, and presentation.

2. A Framework for Management of Concurrent Markup Hierarchies

The framework we propose (Figure 1) generalizes the traditional XML processing framework: parsing an XML document into a DOM data structure (or, alternatively, constructing DOM from an XML database), then using the

DOM API support for editing, querying, and transforming the XML document.

The core of our processing framework is the data structure for storing concurrent XML markup: the GODDAG data structure first introduced by Sperberg-McQueen and Huitfeldt. We enhanced the GODDAG with API and we have designed and implemented a parser for building a GODDAG (Iacob, Dekhtyar & Kaneko) from separate XML files, one file per hierarchy (this would present the advantage of a basic concurrency control over authoring the document encodings). In general, a GODDAG data structure can be built using specialized drivers for different concurrent markup representations.

In Iacob, Dekhtyar & Zhao, we present an extension of the XPath query language for querying concurrent markup represented as a GODDAG. As GODDAG represents a "multidimensional" generalization of DOM, our extension of XPath generalizes XPath to deal with concurrent hierarchies. In the absence of multiple hierarchies, GODDAG reduces to DOM whereas extended XPath reduces to XPath (the extended XPath semantics is given at: <http://dmlab.csr.uky.edu/~eiaco0/docs/expath/>). With the parser and the query language we provide answers to two of the open problems in Sperberg-McQueen & Huitfeldt. Moreover, the presentation issue (XSL) is implicitly solved as we employ patterns expressed in the extended XPath language we propose. The XML editorial tools are based on the GODDAG API: text (PCDATA) update, markup insertion and deletion, and searching (using the XPath extension).

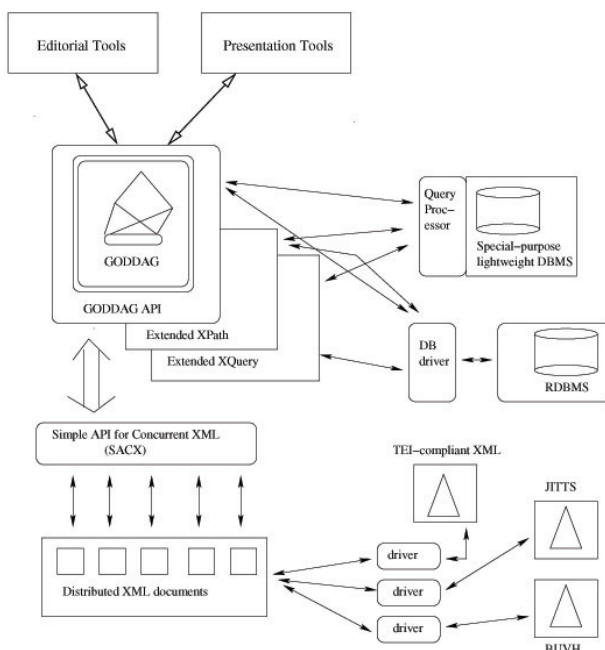


Figure 1: A framework for management of Concurrent XML Hierarchies

For representing and storing concurrent XML markup we defined the notion of *distributed XML document* (Dekhtyar & Iacob): a virtual collection of XML documents, one document per hierarchy. The distributed XML document is obtained via drives from various representations: BUVH and JITTs introduced by Durusau and O'Donnell, XML documents with fragmentation and/or milestones (as in TEI).

Finally, we are currently working on implementing persistent storage support for concurrent XML hierarchies: a specialized database for storing XML with overlapping structures. Our plans include providing support for storing XML with overlapping structures in a relational database.

The framework for processing concurrent XML markup is successfully implemented in the *ARCHway* and *Electronic Boethius* projects (<http://www.rch.uky.edu/>) at the University of Kentucky. The APIs and (part of) the software programs are available at: <http://dmlab.csr.uky.edu/~eiaco0/research/cmh/>.

Bibliography

Bauman, S., A. Bia, L. Burnard, T. Erjavec, J. Hekman, T. Rischer, C. Powell, C. Ruotolo, S. Schreibman, N. Smith, J. Walsh, S. Wells, and F. Wiering (TEI Task Force on SGML to XML Migration). *Strategic Considerations in Migration of TEI Documents from SGML to XML*. Text Encoding Initiative, 2004. Accessed 2005-04-11. <http://www.tei-c.org/Activities/MI/miw02.html>

Czmiel, A. "XML for Overlapping Structures (XfOS) using a non XML Data Mode." Paper delivered at the Joint International Conference of the Association for Humanities Computing and the Association for Literary and Linguistic Computing, June, Göteborg, Sweden 2004. 2004. Accessed 2005-05-25. <http://www.hum.gu.se/allcach2004/AP/html/propl04.html>

Dekhtyar, A., and I.E. Iacob. "A Framework for Management of Concurrent XML Markup." *Special Issue Data and Knowledge Engineering 52* (2005): 185–208.

Durusau, P., and M.B. O'Donnell. "Concurrent Markup for XML Documents." *Proceedings of XML Europe*. Atlanta, Georgia, 2002. Accessed 2005-04-11. http://www.idealliance.org/papers/xml02/dx_xml02/papers/03-03-07/03-03-07.html

Huitfeldt, C. "MECS - A Multi-Element Code System ." 1998. Accessed 2005-04-11. <http://helmer.hit.uib.no/clauser/mecs/mecs.htm>. Forthcoming in Working Papers from the Wittgenstein Archives at the University of Bergen, No 3.

Huitfeldt, C., and C.M. Sperberg-McQueen. "TexMECS: An experimental markup meta-language for complex documents." February 2001.

Iacob, I.E., A. Dekhtyar, and W. Zhao. "XPath Extension for Querying Concurrent XML Markup." *Technical Report TR 394-04*. University of Kentucky Department of Computer Science, February 2004. <<http://www.cs.uky.edu/~dekhtyar/publications/TR394-04.pdf>>

Iacob, I.E., A. Dekhtyar, and K. Kaneko. "Parsing Concurrent XML." *Proceedings of the 6th ACM International Workshop on Web Information and Data Management (WIDM 2004)*, Washington, DC. November 2004.

Sperberg-McQueen, C.M., and Lou Burnard, eds. *Guidelines for Electronic Text Encoding and Interchange*. Chicago and Oxford: TEI P4, 2001.

Sperberg-McQueen, C.M., and C. Huitfeldt. September 2000. Early draft presented at the ACH-ALLC Conference in Charlottesville, June 1999.

Tennison, J., G.T. Nicol, and W. Piez. *Layered Markup and Annotation Language (LMNL)*. Dissertation, University of Wisconsin, 2002. First introduced at the Extreme Markup Languages Conference 2002, Montreal.