Literate Documentation for XML

Kevin M. Reiss *(kreiss@gc.cuny.edu)*
Mina Rees Library
Graduate and University Center
City University of New York

The Current State of XML Documentation

Current practices for the documentation of XML schema have not progressed much beyond the recommendations made in Eve Maler and Jeanne El Andaloussi's (1996) text Developing SGML DTD's: From Text to Model to Markup. That volume recommended an approach to DTD design called document type modeling to SGML DTD designers that, if followed, produces both a reusable and customizable DTD and a well-structured reference manual that documents that DTD.

Despite the development of XML itself, multiple XML schema languages, and the proliferation of XML applications across many disciplines this is still about all the documentation most XML applications provide today. In fact in many cases the documentation provided is much more inconsistent and haphazard than that recommended by Maler and El Andaloussi. The lack of consistent and structured prose documentation is especially problematic to humanists who use XML. To a humanist creating finding aids, marking up historic texts, or preparing a text for linguistic analysis the clear communication of the markup language designer's intentions is critical in making decisions on how to properly apply markup.

The problem of interpreting the designer's intentions is compounded by the fact that XML lacks a formalized, machine-readable knowledge representation technology that allows a designer to explicitly represent in unambiguous fashion the semantics of a XML application (Renear et al., 2002). Given the difficulties that researchers working in this area have made towards realizing a system that can reliably represent machine-readable markup semantics (Dubin et al., 2002)(Dubin et al., 2003) (Dubin, 2003) (Dubin and Birnbaum, 2004)(Marcoux, 2006) it may be more profitable to experiment with a system that can improve the quality of the prose documentation made available with XML applications.

This project proposes a XML schema authoring environment based on the literate programming paradigm that also contains constructs that force the markup language designer to consistently and unambiguously document the application using natural language. This approach can help bridge the gap between current XML documentation practices until tools that can specify machine-readable semantics for XML appear in the future.

Literate Programming and XML

Literate programming directs the author specify both the program source and the prose documentation for that code within the same file (Knuth 1984). This file is then fed through a processor that produces both executable program code and formatted prose documentation for the program. A XML literate programming environment produces both a validating schema and prose documentation for that schema. The experimental XML literate programming system sketched out in this project utilizes the Text Encoding Initiative's One Document Does it All (ODD) literate programming system.

The TEI P5 (Sperberg-McQueen and Burnard, 2005), the most recent revision of the guidelines, has substantially updated the ODD system. The guidelines now use Relax NG for the validating schema component and formally include and document the elements and attributes that make up the ODD system (Burnard and Rahtz, 2004). This new module (Chapter 27) provides TEI users a formal way to document local customizations of the TEI and produce consistent, well-formatted prose documentation for these extensions. The documentation chapter of the TEI P5 describing the ODD states that the ODD is not restricted to just serving as the means to document and generate the TEI, but can also be used to document and produce a schema for any type of markup scheme (Sperberg-McQueen and Burnard, 2005).

The ODD+

This project takes advantage of the general purpose component of the ODD to create an experimental general purpose XML literate programming system that could be termed the ODD+. The support for modularity, element and attribute classes, schema customization, and XML namespaces within the ODD (Burnard and Rahtz, 2004) make it an ideal tool to make up the core of a literate programming system for XML. The advance of the ODD+ system will be the provision of a "semantic checklist" that the author must complete for each element and attribute in a XML application. This checklist will force the markup language designer to document new elements or attributes with precision and consistency. If the checklist is not properly completed the ODD+ processor will fail. The ODD+ application will be implemented by using the TEI P5 extension mechanism to extend the current ODD module to include the elements and attributes that will be necessary to implement the ODD+.

Identifying the Semantic Checklist

What sort of questions will be included in the semantic checklist? The researchers who have investigated the question of XML semantics have identified a number of constructs that occur within markup that seem appropriate candidates for inclusion in the semantic checklist (Renear et al., 2002)(Sperberg-McQueen, Huitfeldt, and Renear, 2000). The questions identified by these researchers include:

1. Can an element have different meanings depending on the content within it is used?
2. Issues of propagation: does the property declared by a given element or attribute apply to child elements, their attributes, and character data contained within them?
3. Issues of class memberships:
    1. Does the element or attribute serve as a superclass?
    2. Is the element or attribute derived from another?
4. Relationships to other elements:
    1. What are the properties that a parent child relationship implies?
    2. What are the properties that a sibling relationship implies?

The author will report on the difficulty of integrating the above questions and others into a workable semantic checklist that can be deployed within a general purpose literate programming system for XML like the ODD. The author will experiment with the ODD+ system and widely used XML applications in the humanities, such as the TEI itself or the Metadata Encoding and Transmission Standard (METS). The author will work with interesting subsets within each application and produce documentation for them using ODD+. This will illustrate the viability of the ODD+ as a potential general purpose literate programming tool for XML.

The author will experiment with different techniques that will strive to ensure consistency of format and language usage in the prose documentation produced by the ODD+. These may come in the form of Schematron rules written to ensure that the author uses language consistency when discusses issues of propagation for example. The author may also experiment with XSLT templates that may contain the outlines of statements that describe class relationships within a XML schema. These outline statements could then be filled in with the specific element or attribute names of the language designer's schema. This approach is an approximation of the skeleton sentence technique suggested by Sperberg-McQueen, Huitfeldt, and Renear (2000). The author will report the results of these various experiments and present a demonstration of the ODD+ system.

Bibliography

Burnard, Lou, and Syd Bauman, eds. *TEI P5: Guidelines for Electronic Text Encoding and Interchange.* TEI Consortium, 2005. <http://www.tei-c.org.uk/P5/Guidelines/index.html>

Burnard, Lou, and Sebastian Rahtz. "RelaxNG with Son of ODD." *Proceedings of Extreme Markup Languages 2004, Montréal, Québec, August 2004.* Ed. B. T. Usdin. Montréal, Canada, 2004.

Dubin, David, and David J. Birnbaum. "Interpretation Beyond Markup." *Proceedings of Extreme Markup Languages 2004, Montréal, Québec, August 2004.* Ed. B. T. Usdin. Montréal, Canada, 2004.

Dubin, David, C. M. Sperberg-McQueen, Allen Renear, and Claus Huitfeldt. "A Logic Programming Environment for Document Semantics and Inference." Paper presented at ALLC/ACH 2002, Tübingen, Germany, July 2002. 2002.

Dubin, David, C. M. Sperberg-McQueen, Allen Renear, and Claus Huitfeldt. "A Logic Programming Environment for Document Semantics and Inference." *Literary & Linguistic Computing* 18.2 (2003): 225-233.

Dubin, David. "Object Mapping for Markup Semantics." *Proceedings of Extreme Markup Languages 2003, Montréal, Québec, August 2003.* Ed. B. T. Usdin. Montréal, Canada, 2003.

Knuth, Donald. "Literate Programming." *The Computer Journal* 27 (1984): 97-111.

Maler, Eve, and Jeanne El Andaloussi. *Developing SGML DTDs: From Text to Model to Markup.* Upper Saddle River, NJ: Prentice Hall PTR, 1996.

Marcoux, Yves. "A Natural-language Approach to Modeling: Why is some XML so Difficult to Write?" *Proceedings of Extreme Markup Languages 2006, Montréal, Québec, August 2006.* Ed. B. T. Usdin. Montréal, Canada, 2006.

Renear, Allen H., David Dubin, C. M.Sperberg-McQueen, and Claus Huitfeldt. "Towards a Semantics for XML Markup." *Proceedings of the 2002 ACM Symposium on Document Engineering, McLean, VA, November 2002.* Ed. R. Furuta, J. I. Malectic and E. Munson. New York: Association for Computing Machinery, 2002. 119-126.

Sperberg-McQueen, C. M., David Dubin, Claus Huitfeldt, and Allen H. Renear. "Drawing Inferences on the Basis of Markup." *Proceedings of Extreme Markup Languages 2002, Montréal, Québec, August 2002.* Ed. B. T. Usdin and S. R. Newcomb. Montréal, Canada, 2002.

Sperberg-McQueen, C. M., Claus Huitfeldt, and Allen H. Renear. "Meaning and Interpretation of Markup." *Markup Languages: Theory and Practice* 2.3 (2000): 215-234.