

# CAS Project Proposal for Summer/Fall 2006

Author: Kevin Reiss

Project Title: Literate Documentation and Document Processing Using XML Schema

## ***Extended Abstract***

XML is a meta-language for defining markup languages. XML allows the user to define a domain specific markup language for a particular application. This markup language is expressed in a machine readable form. When an XML markup language is properly used and designed it adds descriptive markup to a document. Descriptive markup defines the logical parts that exist within a particular document instead using markup to define presentational or procedural information relating to how a document will be displayed.

XML and XML-related technologies currently form the basis of the World Wide Web. Most documents on the web are marked up using the XML-related application HTML. XML applications are very diverse, including Turing-complete programming languages (XSLT), vector graphics (SVG), and container formats for digital object metadata (METS). XML is also in heavy use as a format for message protocols and data-exchange in distributed web applications, which are being increasingly popular for such disparate activities such as metadata harvesting (OAI), and news syndication (RSS/Atom).

XML Schema Languages, including DTDs, Relax NG, and XML Schema, provide a way to formally specify, in machine-readable form, the elements of a given markup language, the patterns that those elements can appear in, and the attributes that those elements can have. XML document instances that are marked up using the elements

of a particular schema can be checked for conformance to that schema. This process is known as validation.

However, XML schema languages still do not satisfy many of the needs of both XML application developers, web programmers utilizing distributed XML data, and encoders. This is because current XML schema languages lack a formal (machine-readable) mechanism for defining the semantic relationships between the elements and attributes in an XML markup language beyond the generic parent-child and sibling relationships that are implicit in the tree data structure that is produced when an XML document is parsed. The ability to represent this would greatly increase the ability of developers to efficiently develop robust tools to process XML document instances marked up using any type of XML application.

Many XML applications also lack adequate and consistent human-readable documentation to facilitate efficient and correct use of the markup language as it has been defined. The documentation for an XML application should represent the markup language in a fashion that is useful to both XML application developers and encoders. While a few XML applications, notably the TEI, have produced well thought-out comprehensive user documentation, most produce only barebones material that is of minimal use. Typically the application developer or encoder is forced to puzzle out the relations between and meaning of the elements and attributes of the language by examining example document instances.

This project seeks to enhance the usability and expressiveness of XML schemas by experimenting with formal mechanisms for describing the semantics of a markup language. A demonstration application will be created, using Donald Knuth's "Literate

Programming” paradigm, that will allow a markup language designer to specify all three aspects important to the effective design and use of a markup language: (1) the machine-readable syntax of the application using an existing XML schema language, (2) natural language documentation about the application for end users, and (3) an experimental means to define the machine-readable semantics of the application’s markup.

The markup semantics will be expressed using a knowledge representation system that is identified as best suited to the task of expressing markup semantics. The KR systems that will be considered as candidates for the means in which to effectively represent these semantics include BECHAMEL, RDF, OWL, and object-oriented modeling tools such as UML.

The literature on the use of XML as a tool for semantic modeling will be reviewed and consideration will be given how the project’s literate programming application, which will attempt to provide the language designer with the capability to express the markup semantics of an XML application, could help redress the inadequacy of XML Schema Languages as information modeling tools. The work practices and methodologies that have been used by large-scale XML encoding projects, such as the TEI or METS-based digital library projects will also be considered. This will give insight regarding the design of an effective general purpose documentation system for XML applications. This literature review and the demonstration application will be the deliverables produced by this project.

## **References**

Dubin , D. and Birnbaum, D. Interpretation beyond markup. In B. T Usdin, editor, *Proceedings of Extreme Markup Languages 2004*, Montreal, Quebec, August 2004.

Coombs, J. H., Renear, A. H., and DeRose, S. J. Markup systems and the future of scholarly text processing. *Communications of the ACM* 30, 11 (1987), 933–947.

Knuth, D. E. Literate programming. *The Computer Journal* 27, 2 (May 1984), 97–111.

Renear, A., Dubin, D., and Sperberg-McQueen, C. M. Towards a semantics for xml markup. In *Proceedings of the 2002 ACM symposium on Document engineering* (2002), ACM Press, pp. 119–126.

Van der Vlist, E. *XML Schema*, 1st ed. O'Reilly and Associates, June 2002.

### ***Important XML and Knowledge Representation Applications to Review***

1. The TEI P5 project
2. The original TEI ODD system and the Pizza Chef
3. The genesis of the new Docbook Relax NG Schema
4. Digital library implementations using METS, with a focus on how the METS Profile has been used to define documentation for a particular class of METS documents.
5. XML literate programming applications
6. The experience of applications such as OAI, RSS, and SRU/W (The XML version of Z39.50) and their use to process distributed XML data and how machine-readable markup semantics might help improve their efficiency and accuracy.
7. BECHAMEL, OWL and RDF as possible tools to represent “markup semantics”
8. DTDs, Relax NG, XML Schema, Schematron
9. The Namespace-based Validation Dispatching Language (NVDL) as tool for XML validation using several types of XML Schema Languages

### ***Project Schedule (Summer – Fall 2006)***

1. Literature Review: August
2. Rough Draft of Paper: September
3. Demonstration Application: October
4. Final Draft of Paper: November
5. Complete Demonstration Application/ CAS Project Presentation: December/January 2006

### ***Possible CAS Project Committee Members***

(The committee must be composed of 3 GSLIS faculty members, who will review the project paper and presentation.)

1. Allen Renear
2. Bryan Heidorn
3. John Unsworth
4. Jerome McDonough