Editorial

A love affair with markup

Michael Seadle

The author

Michael Seadle is Editor of *Library Hi Tech*.

Keywords

Computer languages, Databases, Internet

Abstract

It was not love at first sight when I met my first markup language sometime in the 1980s. But XML is different. It has a rich and flexible tag-set that lets it function as a database. It is also starting to have tools that allow Web-based display with standard browsers. Describing XML is not easy, but four aspects seem particularly important: separation of data; tool development; standards; and preservation.

© MCB University Press · ISSN 0737-8831

It was not love at first sight when I met my first markup language sometime in the 1980s. I believe the language was Script from the University of Waterloo in Canada. It was powerful, but seemed cumbersome to someone already accustomed to the word-processing delights of Wordstar on the Osborne 1. Some years later the Web made its appearance and I renewed my acquaintance with markup languages. I greeted markup as an old acquaintance, but thought of HTML mainly as a pretty face, a way to format documents for screen display. It showed little intellectual substance for data-handling, and seemed like a better match for those who really cared about bit-depth beauty. I flirted mildly with HTML when Meta-tags gave it a way to offer structured bibliographic information. I was definitely not in love.

Then my work brought me in contact with Standard Generalized Markup language (SGML) and the Text Encoding Initiative (TEI) Document Type Definition (DTD). If HTML seemed lightweight, the tag-set for SGML TEI seemed distinctly overweight. TEI had an astonishingly complete tag-set for marking elements in a literary text at a level of complexity that would challenge most database systems, but no Web-browser would ask it to dance. Without special software like Panorama, SGML could not be seen at all. SGML was a matronly wallflower at the Internet Ball, and like a good matron, SGML introduced me to her youngest daughter, XML (eXtensible Markup Language).

XML is different. It has a rich and flexible tag-set that lets it function as a database, but also is starting to have tools that allow Web-based display with standard browsers. This gives it an unusual potential for combining intellectual complexity with easy access. XML is still young and relatively unknown. Nonetheless companies like Microsoft have taken an interest in it, and library utilities like OCLC have adopted XML as the base for its Dublin Core initiative. Browsers like Internet Explorer 5 can render some XML directly. XML has many admirers especially within the digital library community. At the Joint Conference on Digital Libraries (25-27 June 2001) in Roanoke, Virginia, someone referred to XML in virtually every session. Suddenly, it matters.

Describing XML is not easy. The danger lies in leaving out aspects because so much development is going on today. Four aspects seem particularly important:
(1) separation of data;
(2) tool development;
(3) standards; and
(4) preservation.

Separation of data

A key feature of XML is that it separates data from its manipulation and display, which can be handled by XSL (eXtensible Style Language) and a variety of other tools. XML-authors can create new tags, either by modifying an existing DTD, or by writing a new DTD, or even by creating an XML document with no DTD or Schema to validate against. The tags can have any name in any language, and can have complex relationships with each other, including parent-child dependencies and repeating or unique elements. The tags can enclose any type of data, including sentences, paragraphs, random words, numeric data, even bitstreams that represent images, sound, or video. Tags can also have attributes – among the most important of these relate a particular tag to external systems: for example, the "<encodinganalog>" tag in Encoded Archival Description (EAD), which can be used to match EAD tags to MARC (MAchine Readable Cataloging) fields. This means that XML can function as a database, with all of the same kinds of relationships that are possible in a powerful relational database. But it is a database with only data:

- data about the data-relationships within the DTD and the attributes; and
- the data itself enclosed within the tags.

Tool development

XML requires tools precisely because XML files contains only data. Three types of tools are being developed. The first are editing tools. Since XML is just data, any data-entry environment can produce valid XML. It can be created using simple standard editors or ordinary commercial wordprocessors, as well as highly specialized tools that allow continuous validation against a DTD in order to prevent anyone from entering a specialized tag in an inappropriate location. Validation tools can also be run externally, and, of course, multiple editors can be used.

Exchange Rate Politics in Latin AmericaThe second type of tools are for display. In a strict sense, these tools are unnecessary, because XML tags usually give a clear verbal indication of their meaning. It is possible to read a novel marked up in XML TEI by opening it in a wordprocessor. It will not look pretty, and the more detailed the markup, the more tolerance a reader will need. Recently XSL has provided a standard way to display particular fields with particular fonts in a particular place on the computer screen. And server-side programs like Cocoon (see Hancock and Giarlo, this issue) make it possible to display XML on all browsers by rendering it on the fly in HTML.

The third type of tool is for extraction. One is XSLT (XSL-Transformation), which can select particular fields based on tags and if-then logic. More often those who want to manipulate XML still load it (temporarily) into databases, or write simple programs to process it. Indexing tools are also becoming more widely available to give direct access to particular locations in particular files.

Standards

The standards that define XML seem relatively stable. Standards that support shared meanings for particular tags are growing in importance. For example, the Dublin Core (DC) tag set is deliberately limited so that it can function as a simple, practical way of describing digital resources without the cost of full MARC cataloging. But people use DC for a wide range of types of materials, and in a wide variety of ways. Some people enclose whole paragraphs of information inside a tag meant (in the view of some) to give a one or two line title. A DTD can validate the relationship among tags, but the data that exists between the tags has no external validation tool. One person's long, careful, detailed, description may be another's garbage because the verbosity renders it unusable. Contents standards develop slowly, even when growing out of existing standards for related metadata. Yuen *et al.* (this issue) talk about how "information searching becomes more efficient and effective" with proper standards. Broad agreement is fundamental to genuine interoperability between Internet resources.

Preservation

XML aids preservation for two reasons: it is pure data, and the data is mainly in readable ASCII characters. One of the common arguments for microfilm as a preservation medium has been that can be read with nothing more sophisticated than a magnifying glass. In a sense, the same is true for XML. Essentially every existing computer can open essentially any XML file in essentially any file editor. Such file editors are at least as ubiquitous on today's computers as magnifying glasses are in the desks of the over-50 crowd (I never had one in my pre-bifocal youth). The information inside XML files does not vanish with new releases of proprietary software packages. The version migration issues become far more tractable, because they depend only on the stability of XML itself.

Conclusion

End-users on the Internet do not love XML yet. They have an ongoing infatuation with HTML, and may remain with HTML for some time, since it represents the lowest common denominator for access with existing Web-navigation tools. This will change first for people who want to get information from complex databases, because of the interoperability issues that XML helps to address. Will XML become the lingua franca of Internet metadata? It may, in a few years, if the current level of interest among researchers and commercial developers persists.