

NOTICE

Warning Concerning Copyright Restrictions

The copyright law of the United States (Title 17, United States Code) governs the making of photocopies or other reproductions of copyrighted material.

Under certain conditions specified in the law, libraries and archives are authorized to furnish a photocopy or other reproduction. One of these specified conditions is that the photocopy or reproduction is not to be “used for any purpose other than private study, scholarship, or research.” If a user makes a request for, or later uses, a photocopy or reproduction for purposes in excess of “fair use,” that user may be liable for copyright infringement.

This institution reserves the right to refuse to accept a copying order if, in its judgement, fulfillment of the order would involve violation of copyright law.

**SCANNED DOCUMENT
IS BEST COPY AVAILABLE**

XML Applications in the Sciences

Ronald W. Gilmour

ABSTRACT. The Extensible Markup Language (XML) is emerging as a standard format for structured data. Various scientific disciplines are in the process of defining XML document type definitions (DTDs) for communication among specialists. XML applications in the fields of chemistry, molecular biology, and mathematics are described and the relevance of such applications to science librarianship is discussed. [Article copies available for a fee from The Haworth Document Delivery Service: 1-800-342-9678. E-mail address: <getinfo@haworthpressinc.com> Website: <<http://www.HaworthPress.com>> © 2001 by The Haworth Press, Inc. All rights reserved.]

KEYWORDS. XML, CML, MathML, BioML, BSML, scientific communication, molecular biology, mathematics, chemistry

INTRODUCTION

Now another acronym is appearing on the scene, accompanied by a cloud of marketing hype and misconception. (C. M. Sperberg-McQueen, 1998)

The Extensible Markup Language (XML) has indeed generated its share of hype, and more than its share of misconceptions. It has been touted as everything from a panacea for inefficient web searching (e.g., Funke, 1998) to a cure for the curse of perpetual data-conversion (Wilner, 1998). Some see it as another flashy new web-toy, along the lines of JavaScript and applets, while others view it as a tool for

Ronald W. Gilmour, BS and MS in Biology, is Reference Librarian, Science Library, University at Albany, Albany, NY

Science & Technology Libraries, Vol. 19(2) 2001
© 2001 by The Haworth Press, Inc. All rights reserved.

75

serious scholarship which will represent another step toward a "library without walls" (Friedland, 1998).

A host of articles have appeared in the recent past which provide a basic definition of XML and some information on its syntax (e.g., Bryan, 1998; Davies, 1998; Lewis, 1998), so this will be dealt with only briefly here. In the beginning, there was Standard Generalized Markup Language (SGML), a meta-language which is used to define languages for more specific purposes. The most familiar application of SGML is the HyperText Markup Language (HTML), a language designed for the presentation of documents on the World Wide Web. While HTML is admirably suited to the formatting and presentation of documents, it has only very weak capabilities for describing the information within a document. Because information comes in so many specialized forms, a single language for the description of any kind of information would be extremely unwieldy. Hence the usefulness of a meta-language. XML is a subset of SGML that allows users to develop markup languages for their own information needs. Its advantage over SGML is its much greater simplicity. Its advantage over HTML is its flexibility and greater descriptive ability.

As an example, an HTML document describing a copy of Darwin's *Origin of Species* might look like this:

```
<head>
  <title>Origin of Species</title>
</head>

<body>
  <h1><i>On the Origin of Species    <i></h1>
  <p>by Charles Darwin</p>
  <p>London: John Murray, 1866</p>
  <p>4<sup>th</sup> edition, with additions and corrections.</p>
  <p>"Francis Darwin" inscribed on front free endpaper. </p>
</body>
</html>
```

The <p> tags indicate paragraphs, while <h1> refers to a top-level heading and the <i> tags to italic text. This document contains abundant formatting information, but almost nothing regarding the nature

of the information that it contains. In contrast, XML document might look like:

```
<book>
<title>On the Origin of Species</title>
<author>
<first_name>Charles</first_name>
  <last_name>Darwin</last_name>
</author>
<imprint>
  <place>London</place>
  <publisher>John Murray</publisher>
  <date>1866</date>
</imprint>
<edition>4th edition, with additions and corrections.</edition>
<provenance>"Francis Darwin" inscribed on front free
endpaper.</provenance>
</book>
```

This is just the opposite of the HTML case: here, there is no formatting information at all, but the nature of all of the data is defined by tags. Of course, this information is only meaningful in comparison with some standard. The XML model involves the development of standard languages for communication among workers in particular fields or industries. For instance, the airline industry has used a standardized SGML application for some time (Wilner, 1998) and tentative steps are currently being taken within the health care industry for a standardized XML markup language for physicians' reports (e.g., Chueh et al., 1998; Shobowale, 1998; Friedman et al., 1999). The document that describes a specific XML, language is called a document type definition (DTD). The DTD defines the markup elements to be used and how they relate to each other. In the XML example above, for instance, a DTD might specify that title may contain no other elements, while author is allowed to contain first name and last-name elements.

The specificity of XML languages combined with the selective

features of the Extensible Stylesheet Language (XSL), allows simple ASCII files marked up in XML to act like a database from which content can be selectively retrieved based on criteria (see Clark, 1999). (For instance, in a "library" document containing records like the example above, an XSL stylesheet could be written to retrieve all records written by authors with the last name "Darwin" which were published prior to 1900. Furthermore, the stylesheet could specify the exact format of the output, so the records could be ordered alphabetically by title with the publication date in bold type.)

To summarize, then, XML has an immense capability for providing easy access to structured information in an almost database-like manner, but the effective use of this potential is largely dependent on the development of discipline- or industry-specific DTDs. The main goal in the present paper is to discuss how scientists are using XML and what significance these scientific applications have for libraries.

SCIENTIFIC APPLICATIONS

General Principles: Text vs. Data

. . . there is strong similarity at the abstract level between the structures of documents describing a Shakespeare play, taxonomy, an mmCIF [crystallographic information file] and an engineering materials catalogue. (Peter Murray-Rust, 1998)

Before taking a close look at the use of XML in the sciences, it is instructive to see how it has been used in other disciplines. The humanities have a long history with SGML (philosophically close enough to XML to be synonymous in this discussion), especially within the archival community. Encoded Archival Description (EAD) is an SGML language that has become a well-established vehicle for the publication of archival finding aids on the Web. The Dublin Core is a set of elements for resource description (cataloging, in library parlance) which can be used as a basis for more elaborate metadata systems (Weibel, 1997; Dublin Core Metadata Initiative, 1999). It forms the basis of the Resource Description Framework (RDF), a metadata format developed under the auspices of the World Wide Web Consortium (W3C) (Miller, 1998). The Text Encoding Initiative (TEI) is a general standard for encoding any type of text (see Friedland,

1998). (TEI has traditionally been an SGML application, but a “lite” version of the DTD for XML has been informally proposed (Sperberg-McQueen, 1999).) These applications are all similar in that they involve the marking-up of an extant text in such a way that it can be easily searched and manipulated. An extensive listing of TEI projects is available through the TEI website (www.uic.edu/orgs/tei/).

Clearly, textual mark-up is a powerful tool for analysis and a means of adding value to extant documents. There is also a certain familiarity about the process of text-markup to anyone who has used HTML. You begin with a document that you wish to make available on the web and then you add tags to it until you have a webpage. (In the case of XML, of course, a stylesheet would also be needed for effective web presentation.) Arguably, though, this is not the most powerful or characteristic use of XML. A text will have its own inherent structure, which may or may not lend itself to emulation in the strictly nested hierarchy of an XML document. Often, texts may have multiple overlapping structures (e.g., pages and chapters) which cannot both be represented by XML elements.

In contrast to text encoding, scientists are often concerned with using XML as a means of storing, accessing and transmitting raw data. Within this context, XML acts like a low-overhead database with the capability for report-exchange via the web (or any other Internet or intranet application). With data that does not have a pre-defined format, the XML coding can be tailored very precisely to fit the data, without the compromises inherent in text encoding. The following examples demonstrate how this power has been put to work by innovative workers in scientific fields.

Chemistry: CML

... it's no harder than Cahn-Ingold-Prelog chirality rules, and easier than Huckel theory. (Peter Murray-Rust, 1999)

The Chemical Markup Language (CML) was developed by Peter Murray-Rust both for the description of molecules and as a medium for the publication of entire chemical articles. It is supported by its own Java-based browser, Jumbo, also created by Murray-Rust. The CML approach to markup is highly appropriate for a language with ambitions to be adopted by a large and diverse field. Its goal is to provide a relatively small number of tags that may be used in almost

any combination. A look at the CML DTD (CML-1999-0515.dtd, available from the CML website at www.xml-cml.org/, and also reproduced in Murray-Rust and Rzepa, 1999) reveals that there are only 24 element declarations, half of which are completely unrestricted in terms of what they may contain. The philosophy behind this approach is to make it possible for specialists in certain fields to develop conventions as to how these tags should be used, a reasonable compromise between standardization and flexibility.

CML used with Jumbo demonstrates the scalability of browser displays of XML data. An XML file describing a particular chemical might contain abundant information on its physical and chemical properties, its structure, bond angles, formal charges, and so forth, but only the information which the user actually wants to see need be displayed. Complex structural information is treated by using arrays of X and Y coordinates combined with an array of chemical element symbols. Crystallographic measurements may be encoded in a similar way.

In constructing the sample files which come with the Jumbo download, Murray-Rust has taken advantage of the new power which XML grants to entities (best known to HTML users as stand-ins for special characters, such as " for a quotation mark). In XML, entities can refer not only to single characters, but to strings, graphics, or entire files. This means that XML data may be stored in small files which can be pulled together for particular purposes. A researcher working with a certain group of steroids, for instance, could write CML documents describing these compounds and then refer to them in his or her XML-encoded papers with entity references. This makes for small file sizes and minimizes time spent typing, cutting and pasting.

The browser Jumbo renders CML pages as a collapsible hierarchical tree, similar to the directory tree shown in the left-hand pane of Microsoft Windows Explorer. The nodes are distinguished by different symbols indicating the type of information stored in each node. Details of a particular node, such as text or molecular structures, are displayed in a second pane. When the user selects a new node, the contents of that node are opened in a new pane, with a tab control provided for easy reference back to the previous level in the tree. Stylesheets allow for the viewing of molecules and may be customized to control the appearance of certain features (e.g., Should benzene rings appear with an included circle or with alternating single and double bonds?).

It is important that the value of CML and similar languages not be judged by the aesthetics or even the functionality of their custom browsers. Unlike HTML, the emphasis in XML is on data, not presentation. Once there are ways of encoding scientific data in a meaningful, software-independent format, more time will no doubt be devoted to browser support and the integration of this very flexible technology with other kinds of software.

Molecular Biology: BioML & BSML

Information measures the freedom of choice, and thus the improbability of the message; but it is unaware of the semantic content. Any material structure can therefore be compared to a message, since the nature and position of its components, atoms or molecules, are the result of a choice made from a series of possible combinations. (François Jacob, 1970)

Biologists have been wrestling with the problems associated with large volumes of data since the molecular revolution of the 1960s. Currently, there are two major proposals for XML markup of biological sequence data, both from commercial sources: the Bioinformatic Sequence Markup Language™ (BSML) from Visual Genomics, Inc. and the Biopolymer Markup Language (BioML) from Proteometrics, LLC (Beavis, 1999 and Fenyő, 1999). While similar in many respects, the differences between these two DTD proposals highlight some of the philosophical and practical issues associated with data markup.

At first glance, proteins and nucleic acids do not seem to lend themselves to a hierarchical descriptive model. Fundamentally, they are long strings of chemical residues (proteins being composed of amino acids, nucleic acids of nucleotides), but describing these structures in XML provides some surprising benefits:

- Sequence data may be nested within taxonomic elements to provide an organismal context for the molecular-level description. (This feature is best developed in BioML.)
- Sub-units of the biopolymers may be described at any depth of complexity. For instance, in a file describing DNA, genes may be located within certain chromosomes and may contain introns, exons, start and stop codons, binding sites, and so forth.

- Notes and bibliographic citations may be indicated at the level to which they apply. (For example, the tags for a particular protein domain may provide a link to a publication describing it.)

One unusual feature of both languages is that sequence data is provided in one large block, and is annotated by descriptive elements that occur after the sequence itself. So, using BioML as an example, rather than sections of a sequence set off by tags, thus:

```
<peptide>
  <signal_peptide>MSIMGLKVNVS AIFMAVLLT</signal_peptide>
  [LQPTPGQIHWG
  <potential_glycosylation_site>N</potential_glycosylation_site>

  LSKIGVVGIGSASYKVMTRSSHQSLVIKLMPNITLLNN
  <potential_disulfide_linkage>C</potential_disulfide_linkage>
  TRVEIAEYRRL
</peptide>
```

we have:

```
<peptide>
  MSIMGLKVNVS AIFMAVLLTLQPTPGQIHWGNLSKIGVVGIGSASYKVMTRSSHQSLVIKL
  MPNITLLNNTTRVEI AEYRRL
  <domain label="signal peptide" end="20" id="1" start="1"/>
  <group label="Potential glycosylation sites">
    <aa at="32" type="N">
      <amod label="N-glycosyl" at="32"/>
    </aa>
  </group>
  <group label="Potential disulfide linkages">
    <aa at="71" type="C" to="198"/>
```

```
</group>  
</peptide>
```

This seems unwieldy at first, but has the advantage of keeping sequences together for easy transfer into other applications and of allowing the possibility of overlapping sequences, for instance if the results of two studies were merged into a single document.

Both languages are associated with custom browsers that allow users to view the raw sequence data in visually enhanced ways and to access various types of metadata (bibliographic, database links, and so forth). BioML is available for free download, as is the "Basic Browser" for BSML. An enhanced "Browser Plus" version is available for purchase \$500 as of Dec. 8, 1999). Both browsers are very easy to download and to use and notably do not require the Runtime Environments and class libraries needed for the Java-based Jumbo.

Like Jumbo, BioBrow displays the document structure in a left-hand "explorer"-like panel and displays the content of selected nodes on the right. Icons at nodes in the document tree correspond to the type of (data in the node. Proteins appear with optional, user-controlled, visual representations of certain characteristics, such as alpha helices, disulfide bonds, and N-glycosylation sites. The sequence appears flanked by two columns of numbers for easy calculation of positions.

The BSML browser is considerably fancier in terms of its graphic capabilities,, offering users the ability to zoom in on parts of a sequence, to view the sequence in color-coded functional sections or in a scrolling, plain-text sequence viewer, and also providing viewers for the various types of charts and graphs which may accompany publications. There is also a special viewer for displaying electrophoretic gel data.

In terms of the language itself, BioML is more in keeping with the W3C goals of simplicity and separation of format and content (cf. Bray et al., 1998). The BioML DTD is short and simple, with a mere seven element declarations (albeit with a lot of entities). The DTD begins with a prolog area in which changes to the DTD over time are listed, a useful practice for authors of DTDs. The BSML DTD is quite large, with 111 element declarations, many of them of a formatting nature (e.g., "style," "margin," "tables," "fonts"). While technically

an XML application, this emphasis on format makes BSML seem philosophically more HTML-like. Arguably, though, an HTML-like approach to data which is not “human readable” in the first place may be appropriate.

A final point in the comparison of the two systems is that BSML offers the ability to very easily import data from Genbank (the National Center for Biotechnology Information’s online databases of biological sequences, <http://www.ncbi.nlm.nih.gov/Genbank/>) and convert it to a .bsm file for browsing and manipulation in the BSML browser. This will be a major draw since it links this newer technology to one that is well established among biologists.

The presence of two competing languages with incompatible custom browsers raises an interesting point, since XML was designed to be a largely software independent language for free data transfer. It would be sadly ironic if the future of XML actually holds a host of specialized browsers, with data in this “free” format tailored specifically for use on proprietary software.

Mathematics: MathML

Mathematical notation is designed to create the correct ideas in the mind of the reader: It is deliberately ambiguous and incomplete: indeed, it is almost meaningless to all other readers. . . . A formula gets its meaning . . . only when used to communicate between two minds which share a large collection of concepts and assumptions, together with an agreed language for communicating the associated ideas. (Poppelier et al., 1992)

The case of mathematics and XML is an interesting one, since the effort to make mathematics readable on the web antedates XML. This effort began as HTML-Math and has been recast as an XML application. It is also unique in that it is the only “special-interest” web-language application in which the W3C itself has played a major role. In keeping with the W3C’s interest in accessibility, MathML is designed to be easily rendered by aural or visual agents. This is an important consideration since the bulk of the mathematical equations currently on the web are images, which cause problems for voice browsers. Even the use of the “alt” attribute is unlikely to solve the problem if a complex equation is involved. Hence the need for a “native” way of

rendering mathematics on the web. Such a system also introduces the possibility of being able to manipulate mathematical expressions on the web as we currently manipulate text (e.g., copying and pasting between documents or applications).

The most compliant and readily available editor/browser is the W3C's own Amaya. Browser plugins and Java applets are also available for use with MathML. The files created by the Amaya editor are actually normal HTML files conforming to the HTML 4.0 Transitional DTD, but with the MathML expressions enclosed in `<math>` tags, analogous to the `<xml>` tags used for creating XML "data islands" in HTML documents. As a browser, Amaya will not accept XML documents, or rather it will try to render them as HTML.

MathML is actually a sort of double language, with about one quarter of the DTD devoted to tags which describe the formatting of an expression, while the remaining three quarters of the tags provide semantic information about mathematical expressions. This is a difficult division to observe, since the distinction between meaning and notation in mathematics is a very fine one (Ion and Miner, 1999, §1.1). An example of presentational markup would be:

```
<mrow>
  <mfrac>
    <mrow>
      <mn>2</mn>
      <mo>+</mo>
      <mi>x</mi>
    </mrow>
    <mn>3</mn>
  </mfrac>
  <mo>=</mo>
  <mn>4</mn>
</mrow>
```

This represents the expression $(2 + x)/3 = 4$. The same equation might be expressed semantically as:

```

<apply>
  <eq/>
  <cn>4</cn>
  <apply>
    <divide/>
    <cn>3</cn>
    <apply>
      <plus/>
      <cn>2</cn>
      <ci>x</ci>
    </apply>
  </apply>
</apply>

```

Here, the code consists of nested `<apply>` tags, each having as its children a function and two arguments. This type of markup is not meant for graphical rendering, but aims at the expression of mathematical ideas independent of any particular system of notation. Presumably, this would be used primarily for computer-to-computer communication. Note that the semantic markup does not depend on the order of elements (e.g., the positions of lines 9 and 10 in the above example could be switched with no effect on the meaning), while in the presentational markup the order of elements is significant (e.g., if you were to switch the positions of the `mrow` and `mn` elements within the `mfrac`, the fractional portion of the equation would be inverted).

One feature of MathML that is immediately obvious is that it is verbose. A “chalkboard” rendition of the equation above involves 7 figures, while the two forms of markup require 101 and 127 figures, respectively. MathML takes to an extreme the W3C’s advice that “terseness in XML is of minimal importance” (Bray et al., 1998, §1.1) It is designed for use with some form of editorial software and not to be written out by hand.

In order to be acceptable to the mathematics community and to

allow the use of archival data, MathML must be convertible to and from TeX, a system for the computer transmission of mathematical data which has been in use since the late 1970s. In fact, the MathML pages from the W3C currently list a number of products available for converting between the two systems, primarily geared toward the MathML presentational markup.

SIGNIFICANCE FOR LIBRARIES

. . . anything that affects information will also impact the library field. (Exner and Turner, 1998)

XML seems destined to have a large impact on librarianship. At the broadest level, as noted in the above quotation, librarians cannot afford to be ignorant of developments in information technology. This maxim holds true both for public service, where a knowledge of the patron's information habits is an important starting point in providing research assistance, and for technical services, where the use of XML for bibliographic metadata will likely require librarians both to use and author XML documents (see Miller, 1998 and Sperberg-McQueen, 1998).

XML may have further implications in libraries with a large scientific clientele. The greater detail and flexibility that XML offers to web authors may mean that the web will become an increasingly viable alternative or supplement to traditional publishing. As scientific journal prices skyrocket into increasingly ludicrous figures, the academic community can be expected to take greater control of the publication process and to devise independent standards for web publication. Popelier et al. (1992) describe a vision of a publishing utopia in which "all publishers would accept one standard dtd, and all text-processing systems would be able to generate documents prepared according to this dtd, and all bibliographic databases would be able to store this material." Such a system would mean, among other things, an end to idiosyncratic "instructions to authors." This vision is far in the future, but the use of XML for scientific communication is a step in that direction.

The plurality of software required for different XML languages raises issues for libraries' systems departments. Should XML-compatibility be a factor in the choice of browsers to make available on public

terminals? To what degree should non-standard browsers be supported? Should public terminals be Java-enabled? These immediate questions raise the larger issues of how the World Wide Web will change in the coming years. Perhaps the current model of a homogeneous information format (HTML) accessed by a handful of major browsers will shift, with more structured information requiring more advanced browsers or plugins. Jon Bosak (1997) envisions a future in which use of the web will often involve the downloading of XML files followed by their (off-line) manipulation through Java-based interfaces. The "off-line browsing" features incorporated into Microsoft's IE5 seem to be preparing for such a model.

Finally, the use of XML in scholarly communication once again raises the bar on information literacy, which will inevitably impact the user education functions of libraries. The information serving model described by Bosak requires the patron to be more than just a consumer, but to interact intelligently with the information that he or she is accessing. This is not an unprecedented event. Library OPACs and web search engines already require a high degree of interaction from the user, and as with these sources, it will fall to librarians to teach the art and science of their use.

REFERENCES

- Beavis, R. (1999). The Biopolymer Markup Language BioML Working Draft Proposal. Available: http://204.112.55.140/BIOML/b_toc.htm.
- Bosak, J. (1997). XML, Java, and the future of the web. Available: <http://metalab.unc.edu/pub/sun-info/standards/xml/why/xmlapps.htm>.
- Bray, T.; J. Paoli; and C. M. Sperberg-McQueen. (1998). Extensible Markup Language (XML) 1.0. Available: <http://www.w3.org/TR/1998/REC-xml-19980210.html>.
- Bryan, M. (1998). An introduction to the Extensible Markup Language (XML). *Bulletin of the American Society for Information Science* 25(1): 11-14.
- Chueh, H. C.; W. F. Raila; D. A. Berkowitz; and G. O. Barnett (1998). An XML portable chart format. *Proceedings of the American Medical Informatics Association Annual Symposium 1998*: 730-4.
- Clark, J. (1999). XSL Transformations (XSLT) Version 1.0. Available <http://www.w3.org/TR/1999/REC-xslt-19991116>.
- Davies, R. (1998). XML: a new web for the new century. *Feliciter* 44(10): 16-19.
- Dublin Core Metadata Initiative. (1999). The Dublin Core: a simple content description model for electronic resources. Available: <http://purl.oclc.org/docs/core/>.
- Exner, N. and L. F. Turner. (1998). Examining XML: new concepts and possibilities in web authoring. *Computers in Libraries* 18(10): 50-53.
- Fenyő, D. (1999). The Biopolymer Markup Language. *Bioinformatics* 15(4): 339-40.

- Friedland, L. (1998). Text encoding in libraries. Available: <http://www.hti.umich.edu/misc/ssp/workshops/teidl/lfri.html>.
- Friedman, C.; G. Hripcsak; L. Shagina and H. Liu. (1999). Representing information in patient reports using natural language processing and the Extensible Markup Language. *Journal of the American Medical Informatics Association* 6: 76-87.
- Funke, S. (1998). XML content finally arrives on the web. *Searcher* 6(9): 63-68.
- Ion, P. and R. Miner. (1999). Mathematical Markup Language (MathML™) 1.01 Specification. Available: <http://www.w3.org/TR/REC-MathML/>.
- Jacob, F. (1970). *The Logic of Life: a History of Heredity*. (English translation by Betty E. Spillmann, 1973.) Princeton University Press.
- Lewis, J. D. (1998). XML: an introduction. *OCLC Systems & Services* 14(1): 51-52.
- Miller, E. J. (1998). An introduction to the Resource Description Framework. *Bulletin of the American Society for Information Science* 25(1): 15-19.
- Murray-Rust, P. (1998). The globalization of crystallographic knowledge. *Acta Crystallographica D* 54: 1065-1070.
- Murray-Rust, P. (1999). XML-CML.ORG-The Site for Chemical Markup Language. Available: <http://www.xml-cml.org/>.
- Murray-Rust, P. and H. S. Rzepa. (1999). Chemical Markup, XML, and the Worldwide Web. 1. Basic Principles. *Journal of Chemical Information and Computer Sciences* 39: 928-942.
- Poppelier, N. A. F. M.; E. van Herwijnen; and C. A. Rowley. (1992). Standard dtd's and scientific publishing. *Electronic Publishing Special Interest Group (EPSIG) News* 5(3): 10-19.
- Shobowale, G. (1998). SGML, XML and the document-centered approach to electronic medical records. *Bulletin of the American Society for Information Science* 25(1): 7-10.
- Sperberg-McQueen, C. M. (1998). XML and the future of digital libraries. *Journal of Academic Librarianship* 24(4): 314-317.
- Sperberg-McQueen, C. M. (1999). Construction of an XML Version of the TEI DTD. Unpublished document available at <http://www.uic.edu/orgs/tei/ed/edw69.html>.
- Visual Genomics, Inc. Visual Genomics website. Available: <http://www.visualgenomics.com/>.
- Weibel, S. (1997). The Dublin Core: a simple content description model for electronic resources. *Bulletin of the American Society for Information Science* 24(1): 9-11.
- Wilner, E. (1998). Preparing web data with SGML/XML. *Information Today* 15(5): 54, (6): 52-54, (7): 39-42.