1-10-1999

Organizing the Internet

Norman Desmarais
Providence College, normd@providence.edu

Organizing the Internet

by Norman Desmarais

The Internet has sometimes been described as a large library where all the books are scattered on the floor. To get an idea of how difficult it is to manage unstructured data, such as that on the Internet, we just have to consider a search for the word "prince" which will retrieve many more items than we want to examine. It would not be unusual to retrieve thousands or even millions of loosely related URLs on everything from the musician, Prince Edward Island, or Prince Charles (or any one of many other princes) because search engines cannot distinguish between the various definitions of the word: a musician, a part of Canada, or a member of a royal family.

Many people have expended a lot of effort over the years to try to put some order into this chaos but with varying degrees of success. A new development promises to change all that. XML, eXtensible Markup Language, is a new language approved by the World Wide Web Committee (W3C) on February 10, 1998, after about a year and a half of work. Actually, it is more of a meta language, a language for describing languages. It provides the syntax that allows users to create their own markup language and define their own vocabularies to meet specific application or industry needs (hence the "eXtensible" in XML). This capability lets users structure the data in an XML document any way they like and make the descriptions as specific as desired. XML can be used to describe data components, records, and other data structures -- even complex data structures.

This paper intends to examine XML and its relationships with SGML (Standardized General Markup Language) and HTML (HyperText Markup Language). The tags used in any markup language play a crucial role in how other applications understand and work with that data. Document Type Definitions (DTD) identify these tags and their meaning; so we shall consider the XML DTD and proposed alternatives. Not all XML data require a DTD; so we will look at the differences between the two types of XML data: “valid” and “well-formed” documents. Because XML is not a single, fixed format like HTML, we shall discuss some of its “extensions” before considering possible applications and the support it is receiving from software producers.

XML and SGML

XML is a subset of the Standardized General Markup Language (SGML) (ISO 8879:1986 as amended and corrected). Initially conceived for use on the World Wide Web, XML can be used for any type of electronic publication. While SGML is a text processing standard that describes how a document should be laid out and structured, XML is a dialect of SGML that describes the information content of a document.

SGML really didn't catch on very well because it is too complicated, requiring a steep learning curve that corresponds to high costs. People were also reluctant to incur the expenses of hiring a consultant to implement and manage SGML. Instead, they focused on using HTML which, in its pure form, is an application of SGML with a Document Type Definition. But HTML, as it is used in practice, is mostly presentation oriented. It defines how information is displayed, such as the color or font size of a word. It doesn't say anything about the actual meaning of the word. XML, on the other hand, has nothing to do with display. It only describes information.

HTML has a fixed set of tags; but XML lets users define their own tags making it much more flexible and vendor independent. In other words, users can create XML documents in one application and use them in another without requiring a conversion. Because XML accommodates a virtually unlimited number of tags, it can describe the information content of a document more precisely. An XML tag can describe the meaning of any word or term, such as a person's name, a product name, date, or whatever.

For example, searching for the term "pocketbook" in HTML could retrieve items dealing with a purse, a billfold or wallet, or a small book because there has been no way to identify the sense or the meaning of the search term. With XML, developers can tag data to distinguish the word "bill" as a personal name, a bird's beak, a charge, a paper currency or a proposed law -- something which is impossible with HTML.

This is particularly important for numbers which have no inherent meaning in themselves but derive their meaning from the context. While 1000 might be a good price, in dollars, to pay for a new state-of-the-art laptop computer, it would be a very bad average number of days to wait for delivery of purchases. The tag that puts the number in context and gives it meaning becomes critical. The tags also have associations or structure that permit finding, manipulating, acting on, and interacting with the data much more easily.

XML was designed to be easy to implement and to work with both SGML and HTML in such a way that the Web will be able to handle generic SGML in much the same way that is now possible with HTML. XML describes the information content of documents, called XML documents. It also describes to some degree how computer programs will process those documents. XML uses a software module called an XML processor or parser to read XML documents and to provide access to their content and structure. The XML processor interfaces with an application, such as a web browser or word processor, which allows further manipulation of the document. XML permits handling data in a variety of formats and from a variety of sources without costly programming or time delays of data conversion.

Data Type Definition and Alternatives

Those familiar with SGML know that SGML documents use a Document Type Definition (DTD) to specify the structure of a document. The acronym is being extended to Data Type Definition under XML because it is broader and more all-encompassing. Sometimes, the term XML-Data schema is used to avoid any confusion. The XML DTDs or XML-Data schemas can use as data formats the structural relationships between elements and the XML vocabularies which identify those elements used in particular data formats. Only "valid" (more on this later) XML data require a DTD. The DTD is only needed at the time of document creation and not for browsing. It consists of two parts: a formal part for the computer to understand the structure of the information and an informal part for a person to identify what the information is or what it means. It is not envisioned that people will write their own DTDs from scratch but rather use tools like Microstar Software Ltd.'s Near & Far Designer which is a visual modeling tool for building SGML and XML structures that can also convert DTD formats.

The XML-Data schema performs the function of a DTD and specifies the nature of the various elements in a document. The MARC format could easily serve as the XML-Data schema for library applications as it contains all the information needed by librarians and anybody working in the book trade. Catalog librarians might want to distinguish between personal authors, joint authors, corporate authors, etc. and title proper, collective title, uniform title, etc. while book dealers putting together a catalog or students preparing a bibliography are just interested in author and title information without any finer distinctions.

The MARC format as an XML-Data schema could serve all of these functions by describing the syntax for this purpose or for exporting information about books, titles, and authors to a relational database where another XML-Data schema could describe row types and key relationships. XML-Data schemas can provide a common vocabulary for ideas which overlap between syntactic, database, and conceptual schemas making all features usable together as appropriate.

Using the MARC format as an XML-Data schema can also serve to specify books by, rather than about, Charles Darwin, for example. Current engines for searching the Web cannot make this distinction because the document structures do not make it. So they mix both types of books together. Because every database describes its data differently, a search engine cannot understand the schema of each database which describes how it is built. XML could easily categorize books in a standard way using the MARC format so that agents could search bookstore sites by author, title, ISBN, or other criteria in a consistent way, much like Z39.50 engines do for library catalogs. A searcher could identify only stores that stock a particular title instead of retrieving all the places which cite or mention it; or one could limit the search only to articles that review that book.

Logos Research Systems, Inc. developed a small utility that "round-trips" MARC records to "well-formed" XML and back. This utility, called MARCXML, uses a very simple XML format that isn't designed to contain any knowledge of the MARC records other than the basic structure. In other words, it doesn't know what's allowed to repeat, what's obsolete, etc. There is a small DTD; but the converter doesn't require it to rebuild the record. The resulting XML file can be viewed in a browser or edited in a word processor and then recompiled back into a structurally valid MARC record. The XML parser is built into the converter; so the whole thing runs as one small program. A demonstration of the conversion in both directions can be run from Logos Research

Systems's web site at http://www.logos.com/marc. The company also has a free MARCType Parser which parses USMARC records into an annotated human-readable document online.

NICEM (National Information Center for Educational Media) has converted its database from a simple relational database management system to an XMŁ based intranet system in an effort to provide higher-quality new records. By storing the bibliographic records as XML documents, the data are useable by any system that meets standardized criteria.

NICEM has also developed a machine-aided indexing (MAI) application to accompany the new system. The MAI program utilizes the NICEM thesaurus of 4300 subject terms along with 6000 syntax rules written by NICEM editors. It uses natural language processing to read the bibliographic record and then suggest subject terms to the editor. This process takes only seconds and provides the editor with terms they can decide to use or ignore. The editor can also select subject terms from the thesaurus to include in the record. MAI provides for faster indexing and promotes consistency in indexing from editor to editor.

The Graphical Communication Association Research Institute and the Data Interchange Standards Association plan to build libraries of Document Type Definitions, Java applets, template scripts, forms, and object definitions to allow businesses to process the components of XML messages.

Microsoft Corp. and IBM have also developed Document Content Descriptions (DCD), written entirely in XML, to replace DTDs. They submitted the proposal to the World Wide Web Consortium (W3C) in August. The W3C will form a working group to consider it and to flesh out the protocol. DCDs will provide similar functionality to DTDs; but they won't require users to learn XML and a DTD protocol. They will also support other XML-friendly standards such as:

- Resource Description Framework which provides an infrastructure to support metadata across Web-based activities and can be used to describe the content and content relationships available at a particular Web site, page, or digital library;

- XML-Data which standardizes schemas to define the characteristics of classes of objects; and

- XML Namespaces which qualifies tag names used in XML documents by associating them with their source and provides a simple method for qualifying certain names used in XML documents by associating them with namespaces.

- Xpointer (XML Pointer Language) supports addressing into the internal structure of XML documents. This allows going directly to a specific section, paragraph, table, or other element contained in a document instead of referencing the document as a whole.

"Valid" and "Well-Formed" XML

Consistency is an important part of XML implementation. Each type of industry will have to come to some agreement on the metatags its members will use. If we are to be able to compare book prices, for example, booksellers will need to use the same tags. Using synonymous terms like price, list price, selling price, retail price, wholesale price, discount price, or charge to describe the same entity will only perpetuate the confusion and hinder data sharing. In the chaotic world of the Web, it may be difficult to achieve that level of agreement.

In the world of XML and SGML, the DTD specifies the tags and their meanings so applications can work with and manipulate the data. Documents that make use of internal or external Data Type Definition files are known as "valid" XML and require an XML parser to check incoming data against the rules defined in the DTD to verify that the data were structured correctly.

A developer usually decides which parser to integrate into an application or builds one from scratch; so an end-user has no control over which parser to use. There are two kinds of parsers: validating and non-validating. A validating parser “validates” documents by ensuring that they conform to the DTDs specified within them. A non-validating parser only checks that the document has a sound logical structure and syntax ("well formed"). Internet Explorer 4 uses a non-validating parser; but IE5 beta-1 uses a validating parser. Netscape Navigator 5.0 will use a non-validating parser written in C which is faster and more efficient than a Java parser.

Parsers must know what the elements mean and how to process them. XML allows developers to describe element names in a recognizable manner to avoid conflicts between elements with the same name. For example, some applications may have valid reasons for distinguishing between list price, retail price, wholesale price, and discount price, depending on a company’s business rules and markets. Using different tags just to be different, however, will only lead to confusion. The W3C is considering a proposal to make every element name subordinate to a universal resource identifier (URI) to ensure that names remain unambiguous even if chosen by multiple authors.

The creators of the XML format admit that it is "primarily intended to meet the requirements of large-scale Web content providers." However, valid XML files can be used outside the Web, such as in an SGML environment, for electronic publishing or as a means to produce print products.

Data sent without a DTD is known as "well-formed;" and the data can be used implicitly to describe itself. XML encoded data is self-describing and can also be used to model and deliver structured data without any reference to documents. This allows XML's open and flexible format to exchange or transfer information. It also makes XML extremely powerful, as it can be used for any type of information, such as articles, chapters, books, maps, weather forecasts, invoices, purchase orders, and order forms.

XML only requires data to be "well-formed." In other words, its logical structure must appear sound. Its elements must be properly nested with all opening tags subsequently closed. Otherwise, browsers will look for a closing tag until it times out, resulting in slower speed of delivery to end-users.

Extensions

The W3C established the SGML Editorial Review Board in 1996 to develop what would become the XML standard. This board became the XML Working Group. Their goals in designing XML were to make it compatible with SGML, easily transferred over a network, capable of supporting a wide variety of applications, and easy to write programs which process XML documents. They also wanted to minimize the number of optional features in XML and to make XML documents human-legible, reasonably clear, easy to create, and consistently processed by the receiver.

The current XML specification is only a starting point, however. It describes the core language that specifies how to define XML documents and establishes the grammar that make documents comply with the XML specification. Two extensions have been proposed to extend XML beyond the core language: XML Linking (XLL), which may be renamed XLink, and Extensible Stylesheet Language (XSL).

XLL is an XML linking language that describes unidirectional and multi-ended links between objects using XML syntax. It provides links in XML similar to those in HTML; but it is more powerful. It allows for multidirectional linking and links that exist at an object level (e.g. paragraph or section) rather than just at a page level. It is designed to enhance the hypertext links that make the Web work. Its SGML parent is HyTime.

Display issues such as font, justification, color, etc. will be handled by style sheets. XSL, which is still a working draft at the time of this writing, will provide the display semantics for XML. It will map XML into HTML or any other formatting language. It will be based on the SGML style sheet standard, Document Style Semantics Specification Language (DSSSL); but it will use XML syntax rather than DSSSL. It will be compatible with Cascading Style Sheets (CSS), the HTML style-sheet standard, thereby permitting easy and accurate translation. DSSSL attempted to define rules for displaying SGML-tagged data and to provide a superset of the Cascading Style Sheets (CSS) functionality. XSL will be able to generate CSS along with HTML and handle the translation of XML-based data to HTML or other presentation formats.

Style sheets will usually be kept in external files and referenced from within an XML file. XSL will also allow developers to build a presentation structure that differs from the original data structure because the data and its presentation style will be two separate entities. This will allow developers to format and display data elements in multiple places on a page, rearrange them, or remove them. For example, the same bibliographic data source could be used to create a catalog, a bibliography, a footnote, a citation, a purchase order, or an invoice merely by changing the style sheet and without changing the data.

Application Areas

XML will be particularly useful for enabling a Web client to interact with two or more sources of data at the same time. These sources may have different formats or require access through dissimilar interfaces. If they have common tags giving meaning to the data attributes, XML applications will be able to work with them. Search engines will be able to understand and use contextual information when performing a full-text search, producing more relevant results.

XML can take the strain off Web sites by moving much of the processing load from the Web server to the Web client where a Java applet downloaded to the user's PC, for example, may do the processing. Dynamic HTML also uses this approach; but an applet can do more with XML-tagged content because the data possess much more meaning. This will have a dramatic impact on Electronic Data Interchange (EDI) and electronic commerce.

A Web client can interact with multiple views of the same data, unlike in the HTML world where each different view requires a trip back to the server for re-presentation. An XML client can sort, filter, and manipulate data locally, making it much easier to customize a presentation to the needs of the person using the data. A catalog librarian can have one view of the data; a library patron can have another; accounts payable can have yet another. By supporting a greater granularity of information, XML allows individuals to extract relevant information from several sources and reassemble it into any format (data repurposing).

Intelligent Web agents can also be programmed to find and deliver information to meet profiles tailored to each individual. Those profiles could be customized in terms of both content and presentation. After retrieving the information, the browser could use the tags to omit irrelevant portions and display only those portions of a document most likely to interest to the reader. This could reduce information overload and accelerate learning. This capability would also allow individuals to create “custom” documents or reports with the most recent information formatted and delivered any way they want it.

Support

Microsoft (http://www.microsoft.com) is working with the W3C XML Working Group and other companies to help develop the XML standard; ensure interoperability on multiple systems and browsers; and support developers, authors, and users. The company offers XML Notepad to create XML files quickly without hand coding. The existing version is free; but it only creates "well formed" (syntactically correct) XML documents. It does not yet support creating or editing "valid" documents. Microsoft is working with DataChannel, Inc. (http://www.datachannel.com/) to produce an XML parser written in

Java which will enable the integration of a variety of data and applications.

Microsoft's Internet Explorer 4.0 already supports some XML applications; and Explorer 5.0 and Netscape's Navigator 5.0 will both support XML fully. Microsoft has also announced that the next version of Office will use XML as a native file format to maintain formatting while moving documents between Office and the Web. Users will be able to switch back and forth between data in an Excel spreadsheet, for example, and the same data in HTML tables in a Web browser.

ArborText (http://www.arbortext.com/) ADEPT v.7 also reads and writes native XML and can automatically convert documents between XML and SGML. The company also produces XML Styler for creating and modifying XSL stylesheets. In addition to ArborText, BackWeb Technologies, Inc. (http://www.backweb.com), Chrystal Software (http://www.chrystal.com/), Inso (http://www.inso.com/), OmniMark Technologies (www.omnimark.com), POET Software Corp. (http://www.poet.com/), Microstar Software (http://www.microstar.com/), Texcel Research, Inc. (http://www.texcel.no), and webMethods (http://www.webmethods.com) produce tools for authoring, editing, and storing XML documents or generating DTDs. Aeneid, Allaire Corp., ExperTelligence Inc., InterMax Solutions Inc., Pictorius Inc., Powersoft Corp., and SoftQuad Inc. have also committed to providing XML support in their products. Software producers such as Interleaf (http://www.interleaf.com), Marimba, Webcasters, PointCast. DataChannel Inc. (http://www.datachannel.com/), Chrystal Software, UserLand Software (http://www.scripting.com/frontier5/xml), POET Software Corp., and Vignette Corp. (http://www.vignette.com/) have tools for database publishing, content management, and data management.

Because XML is slimmed-down SGML, we might suppose that existing SGML publishing systems would work with XML without any modifications. This is not the case, however. XML's simplifications and different way of interpreting white space require developers to tweak the code. Those who don’t want to mess with the code can use authoring tools like Microstar Software Ltd.’s Near & Far Designer that support conversion to and from various DTD formats and have drag-and-drop features that let users copy structures in one DTD and paste them into another. XML is both vendor-neutral and media-independent; but, because it was designed to accommodate all world languages, it is case sensitive.

Prognosis

XML is not a single, fixed format like HTML, nor is it a replacement for HTML. XML is a metalanguage that lets users design their own markup languages to meet specific application or industry needs. It provides a standard way for describing and exchanging data regardless of its nature, how the sending system stored it, or how the receiving system will use it.

HTML will continue in use for the foreseeable future because of the cost and labor to tag existing documents. The change will be evolutionary; but its impact will be revolutionary. Versions of HTML subsequent to HTML 4.0 will use XML syntax and support XML tags. As Web pages get updated and modified, the code will gradually get converted to XML, producing a richer, more relevant, and better organized Web.