XML: the answer to everything?

This article weighs the pros and cons of XML for some applications (publishing), and explores why it is the best possible solution for many programming and publishing needs.

Kay Ethier Scott Abel

This article weighs the pros and cons of XML for some applications (publishing), and explores why it is the best possible solution for many programming and publishing needs.

Everywhere you turn these days, someone is talking Extensible Markup Language (XML). Jump into a discussion about publishing - XML is touted as a means of exchanging information. Talk with someone about the new software tool she is creating - she describes setting up some of her actions in XML. Ask a webmaster what he's been doing - he raves about the dynamic content he's serving up to site visitors using XML from a database. In short, XML is a great solution to a wide variety of challenges, and it seems to be everywhere. But is it the cure for every data or content challenge? The simple answer is, no.

> Everywhere you turn these days, someone is talking Extensible Markup Language (XML)

Not everyone needs XML to make things work. For some small organizations, publishing processes are straightforward enough that the costs of implementing an XML solution may not be worthwhile. But the only way to be sure is to perform a thorough examination of the business processes and review cycles that produce information products - most organizations and companies grossly underestimate the amount of information they could potentially reuse in publishing, and overestimate the costs of reusing that information with an XML-based solution. And they're not aware of the breadth of available free tools that can get them well on the road to their XML destination.

What is XML?

XML is meta markup language that is used to create new markup languages. It's most commonly used to create tag sets and processing instructions that describe structured content for presentation in text documents, but it can also be used to describe, manage, and deliver content of all types (text, images, voice, forms, multimedia files, and so on) and to transform transactional data between disparate database systems.

> XML is meta markup language that is used to create new markup languages.

Unlike Hypertext Markup Language (HTML), which is a display markup language with a predefined list of tag sets designed solely to control how information is presented in a web browser, XML presents content in an open, standards-based, media-neutral, operating system-agnostic, platform-independent format. XML is extensible because it allows organizations to define their own sets of tags, each with a meaningful (semantic) "name". Semantic names (or tags) are more useful than generic HTML tags because they can describe content in real-world, user-friendly and context-specific ways. For instance, the XML tag `<product name>` is much more descriptive than the HTML tag `<h2>`.

In a traditional word processing environment, the formatting data is stored with the content it governs, and changes to the formatting involve changes to the content itself. XML's strength comes in its ability to separate content from formatting data, thus allowing authors to create content without spending unnecessary time formatting that information. XML style sheets control the formatting of the content being created, and specify how it will be presented in each medium.

XML content can therefore be automatically transformed (with the help of style sheets) from a single text source into a variety of information products (printed product brochures, web site content, wireless content, etc.) each with its own look and feel. And, XML content can be personalized and delivered dynamically on the fly, based on the specific requirements of the end user.

XML also differs from HTML in that it allows documentation to be processed by computer software programs, thus allowing organizations to reuse content from disparate data repositories, and recombine that data in ways – and in various media - not possible with HTML. XML supports single source content reuse, and allows organizations to make changes to a content element (like a product description) and have those changes reflected instantly and automatically in every information product that uses that information, regardless of the medium. This ability to reuse information and to make changes once and have them appear globally saves organizations considerable time and money revising, updating, and translating content.

XML content is also "validated" against document guidelines encoded in a Document Type Definition (DTD) and can enforce standards on the authors who develop content. This ability is particularly useful in validated or regulated environments (life sciences companies, legal firms, automobile and aerospace industries, the financial sector) in which completeness, consistent structure, and accuracy of information are all essential, if costly regulatory compliance and legal issues are to be avoided.

W3C Goals for XML

After the world wide web explosion, web users were inundated with miles of good and bad HTML, and the W3C sought a better solution for publishing, cataloguing, locating, retrieving and archiving data. The guidelines they set for this "something better than HTML" resulted in the development of XML. The "design goals" for XML, which set it aside from HTML, include the following (source: W3C (`http://www.w3c.com`)).

1. XML shall be straightforwardly usable over the Internet.
2. XML shall support a wide variety of applications.
3. XML shall be compatible with SGML.
4. It shall be easy to write programs, which process XML documents.
5. The number of optional features in XML is to be kept to the absolute minimum, ideally zero.
6. XML documents should be human-legible and reasonably clear.
7. The XML design should be prepared quickly.
8. The design of XML shall be formal and concise.
9. XML documents shall be easy to create.
10. Terseness in XML markup is of minimal importance.

This article focuses primarily on the second W3C requirement for XML, that it plays well with a variety of tools that perform various tasks. And since the potential uses of XML are countless, and space is limited, we've restricted our scope to the use of XML in publishing.

XML Uses

In the publishing arena, XML is used by authoring and content management tools. Authors use the XML elements and attributes to produce documents. Content management tools use the XML elements and attributes as data that can be retrieved or marked for reuse.

Is this the answer to everything? Well, in the publishing world the answer is sometimes "no", because affordable publishing can sometimes be accomplished without the help of XML - XML would be overkill. However, XML often is the best option for organizations that take the time to evaluate their content lifecycle and to examine how much it costs to create, maintain, translate, deliver, store, reuse, archive, and retire content. A recent study by ZapThink ("XML in the Content Lifecycle Foundation Report Creating, Managing, Publishing, Syndicating, and Protecting Content with XML") found that the biggest - and most expensive - challenge for most organizations today is content reuse. The study found that "Producers of content in the enterprise spend over 60% of their time locating, formatting, and structuring content and just 40% of their time actually creating it." (Source: ZapThink (`http://www.zapthink.com/report.html?id=ZTR-CL100`))

The sad fact is, most organizations don't know how much their content creation and management efforts cost them, and so they assume that XML is not for them. The reality is that the only way to know whether XML is the right choice for your organization's publishing needs is to seek the assistance of a content management expert who can perform an organizational needs analysis, a content lifecycle analysis, and an audit of your existing content. Additional services offered by content management consultants include customer needs analysis, tools recommendations and assistance calculating return on investment. Analysis often identifies obstacles to change (tools, processes, and people) that will need to be addressed before you adopt XML as a publishing solution. Once you know how much it costs, and what obstacles you'll face, you can make an informed business decision about whether to move to XML publishing or not.

XML does provide a lot of options. Exchanging content, for example, is often easier and more affordable with XML than it is with proprietary tools like Microsoft Word. Rather than saving content in a proprietary format, authors can output their document content into XML and pass it along to colleagues or customers who need the content but who may use other authoring and publishing tools. Additionally, XML makes reuse of information easier since formatting data is separated from XML content. Separating content from format is one of the biggest productivity gains an organization can obtain by adopting XML.

> Exchanging content, for example, is often easier and more affordable with XML than it is with proprietary tools like Microsoft Word

XML content may be used to produce one document, and that same XML content can then be harnessed to create additional documents, each with a completely different look and feel. Alternatively, the same XML content can be dynamically served up to various audiences in different chunks or in different sequences using other technologies (see "XSLT", below). This represents a degree of flexibility that HTML simply doesn't offer.

Free XML Authoring Tools

There are a wide variety of free XML authoring tools available for download on the internet. Each has its own strengths and weaknesses, and no one free tool does it all (i.e. your mileage may vary).

Check them out and learn as much as you can about XML authoring before you decide to employ any particular tool:

- Altova Authentic (`http://www.altova.com/products_doc.html`)
- XML Cooktop (`http://www.xmlcooktop.com/`)
- Open XML Editor (`http://www.philo.de/xmledit/`)
- Xray2 (`http://architag.com/xray/`)

XML-related Technologies

Jonathan Robie (`http://www.gca.org/papers/xmleurope2001/papers/bio/s13-1auth2.html`), an XML Research Specialist at Software AG, once exclaimed, "XML doesn't do anything!" In its purest sense, this is true; by itself, XML will not magically repurpose content for multiple media or audiences. XML doesn't provide formatting in the absence of additional technologies. In order to make XML "look good", or turn it into a final deliverable, some assistance from format-conscious technologies is required... but on the other hand, no amount of such formatting technology can turn ugly-duckling HTML content into a coterie of media swans.

XSL and XSLT

In the HTML world, Cascading Style Sheets ("CSS" files) make HTML display as desired... in a web browser. Because XML separates content from its formatting data, you must employ additional technologies to format XML, allowing it to display as you wish. XML can be formatted a few different ways. You can bring XML content into XML-based tools to change its appearance. (You can also use HTML to format XML.) The XML formatting and transforming language (Extensible Stylesheet Language, Transform, "XSLT" for short) can adjust XML output for various display purposes. When you have multiple media in which you want to present your content, XML is far more flexible than its HTML ancestors.

XSLT uses the tags within an XML document to control formatted output. Formatting XML content can be as simple as adding bold to a <`companyname`> tagged object. The formatting can be as complex as telling all of the pieces of an invoice, for example, to display in a certain font, point size, style, etc. in a table and make the table content "sortable" by any of the tags used in your XML content.

Free software tools used for XSLT include Saxon and Xalan (and others). Each allows you to perform transforms without moving your XML content into a proprietary tool that will "trap" you into using that tool in future.

Saxon, created by Michael Kay, is available in several flavors. The "lite" version allows you to do transformations on any PC running the Java Runtime Environment (JRE).

Saxon is available via Michael Kay's SourceForge website (http://saxon.sourceforge.net/)The JRE is available from multiple sites, including java.com (http://www.java.com/en/download/windows_automatic.jsp).

Xalan is an XSLT processor designed to transform XML documents into HTML, text, or other XML document types and is available via The Apache XML Project (http://xml.apache.org/xalan-j/) among other sites.

Free software tools used for XSLT include Saxon and Xalan (and others). Each allows you to perform transforms without moving your XML content into a proprietary tool that will "trap" you into using that tool in future

A good resource for more information on working with XSLT and XML is Mitch Amiano's free software collection, the "Agile Markup Toolkit", which is available at no cost. The CD itself contains several dozen free software installations and links. Any software on the CD also includes reference information that indicates where it came from, allowing you to update as new releases become available. Mitch is a big user of free software, very involved in the free software community, and is also a user of the tools he has gathered on this CD.

Visit the Agile Markup Toolkit's web site (http://home.agilemarkup.com/index.php?option=content\&task=view\&id=55\&Itemid=29) for more information about "Agile Markup Toolkit".

XSL-FO

Another subset of XSL is XSL-FO. The FO stands for "formatting objects." XSL-FO provides a means for formatting XML for presentation. More information on its capabilities is available at the W3C website (http://www.w3.org/TR/xsl/).

XQuery

Some companies may be publishing information stored in a database or even stored as XML. XQuery allows you to query XML, similar to the way SQL is used to access databases. More information, and a great overview, are available from Data Direct Technologies (http://www.datadirect.com/techzone/xml/basics/basics/index.ssp).

XML Performance

How has XML met with the W3C expectations (http://www.w3.org)? Certainly there are many XML-driven websites. Check out Safari, CNN, Fidelity, and Wired, among others. These are dynamically generated pages with XML behind the scenes. At Fidelity, XML ties together web and back-end systems to deliver hundreds of thousands of transaction per hour to its web site customers. Fidelity says it's realizing millions of dollars of savings in infrastructure and development costs by eliminating the need for transformation of data between the company's disparate database systems and by reducing (by 50%) the number of web application servers through which customer data travels. (Source: InternetWeek (http://www.internetweek.com/newslead01/lead080601.htm)).

In publishing, XML has proven beneficial for creating materials derived from information stored in a database or publishing information that developers have created in XML. Some tools can open the XML and style it, providing paragraph formatting along with page layout (and in publishing, presentation is everything!). Such tools, which can automatically style XML, make publishing data easier and more affordable than traditional publishing methods.

However, XML can slow performance, if not integrated properly and appropriately planned for. "Research by IBM Labs shows that even small XML-based documents can increase the CPU cost of a relational database transaction by up to 10 times in the absence of a dedicated XML processing engine. The research concluded that XML parsing could have a 'potentially fatal impact' on high-performance, transaction-oriented database applications that use XML." (Source: nwfusion.com (http://www.nwfusion.com/news/2004/0503xmlaccel.html)). Hardware vendors are rushing to develop new gigabit-speed silicon to address the spread of XML and the processing problems it can sometime cause.

Again, it's important to employ a content management expert with experience in planning and implementing XML solutions before you adopt XML in your organization. XML is a business solution, not an IT solution. Employ it only after developing and conducting a thorough analysis of your organizational business needs, the needs of your customers, and after evaluating your content lifecycle. The results should yield a unified strategy for XML use across your enterprise that will provide measurable benefits and a positive return on investment.

Conclusion

XML isn't the universal panacea... but it is often preferable to alternatives. Particularly in publishing applications, which represent so many ways data can be caught up in proprietary systems, it's a good idea to use non-proprietary technologies for content authoring, management and delivery, and it's crucial to assess and quantify the potential paybacks of XML versus HTML systems.

Copyright information

About the author

Kay Ethier is an Adobe Certified Expert in FrameMaker 7.x and several prior versions. She instructs training classes, performs consulting, and provides support to clients in a variety of industries. Kay resides in the Research Triangle Park area of North Carolina and works for Bright Path Solutions (http://www.travelthepath.com). In 2001, Kay co-authored the book *XML Weekend Crash Course* (Wiley/HungryMinds). She has most recently been a contributing author on *Advanced FrameMaker* (TIPS Technical Publishing) and *XML and FrameMaker* (Apress).

Scott Abel is a technical writing specialist and content management strategist whose strengths lie in helping organizations improve the way they author, maintain, publish and archive their information assets. Scott has considerable experience in XML-based enterprise content management strategy, information architecture, software tool selection, high tech marketing communications, business process analysis, technical writing and editing, usability, content localization, and designing single source solutions.