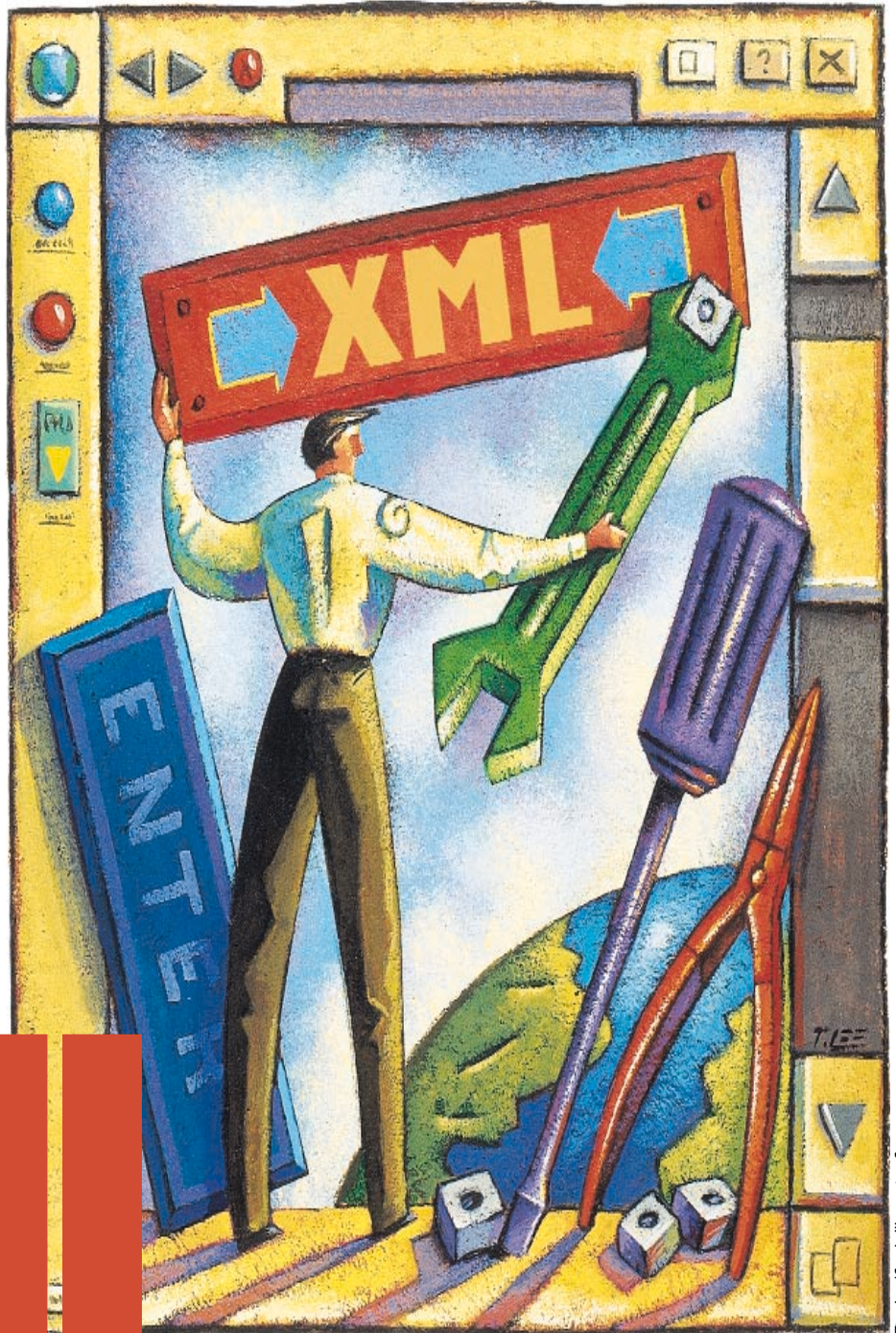


Delivering information online used to be pretty basic. If your company had a hardcopy manual for its product, you likely converted it to some online format, like PDF, *WinHelp*, or *HTML Help*. If you continued to deliver the printed manual, you probably shared content between the printed and the online versions of the documentation. You might have found that reusing your information was tedious—perhaps you had to manually copy and reformat information between the print source material and the online source material.

BY ALAN R. HOUSER
Senior Member, Pittsburgh Chapter



Tim Lee/Stock Illustration Source

Trends in XML Software

Then came the ongoing task of keeping both versions up to date.

Recently, requests to reuse your information may have been growing in number and complexity. Your customer service department wants to create a database of troubleshooting procedures aligned to each product the company sells. Your sales staff wants to provide individualized online documentation that reflects exactly the products and options that each customer ordered. Your field support staff wants hardware repair procedures on their micro-notebook computer screens, as well as a list of all required parts for each procedure. They add, "By the way, it would be really great if we could order a replacement part by clicking on a part number." It's clear that the tools and processes you're currently using won't support these requests.

Enter XML

Today, your content is no longer just static words on a printed page or computer screen; it is being reused in more ways every day. More people want different pieces of your documentation, delivered in a different manner, for different uses in different forms. To accommodate these growing demands on documentation, many companies and industries are moving to XML-based publishing systems.

Moving to an XML-based publishing system is expensive and time-consuming. If your company is taking the plunge, you need to educate yourself about the authoring tools that will allow your writers to create XML. Questions you must ask include the following:

- What can you expect from the current crop of XML authoring tools?
- What additional features can you expect during the coming year?
- How might XML change the way you work?

Why Publish with XML?

Before we look at XML authoring tools, let's do a quick overview of XML. XML is not itself a language, but a syntax for specifying a language. An XML

document typically resembles an HTML document, with elements and attributes surrounded by tags, like <p>. One major difference between XML and HTML lies in the flexibility of the tag names. HTML specifies a set of legal tag names associated with each element, such as
, <p>, and <table>. In XML, however, you develop your own system of tag names to represent the *meaning* of your content. These tags are included in your XML source files. For example, a part number in an XML document might look like this: <partNumber>145-245</partNumber>. These tags provide information *about* the information in your document: "This information is a part number."

This type of information is known as *meta-data*, literally *data about data*. What can you do with this meta-data? You can certainly use it to format your information in print, for a Web browser, for a personal digital assistant (PDA), or even for devices that do not yet exist. Furthermore, you can use it to *select* and *manipulate* the information you need to provide in a given application. Your technical support staff may need only procedures. Your field technicians may need part numbers (perhaps with links to price and

availability information). The possibilities are limited only by your organization's imagination.

Nontraditional Publishing

Authors have traditionally considered presentation and format for print and, more recently, for online presentation on desktop computers, cell phones, and other electronic reading devices. Today, online presentation can be accomplished through a proprietary help interface, a Web browser, or Adobe *Acrobat*, but we expect the number and complexity of output devices to grow. Five years ago, who would have imagined the popularity of PDAs, or that traffic and weather information would be available on the display screen of a cell phone? Who would have imagined the growing popularity of electronic books?

Further, output devices are not limited to the print and online displays that we traditionally recognize. Consider devices that "read" text, converting it to speech. Not only do these devices allow visually impaired people to access printed information, they may also deliver content in environments in which reading is unwise (such as while driving a vehicle) or difficult (such as in hazardous industrial environments).

Figure 1. FrameMaker+SGML's user interface, which shows both a WYSIWYG document view and a structure view.

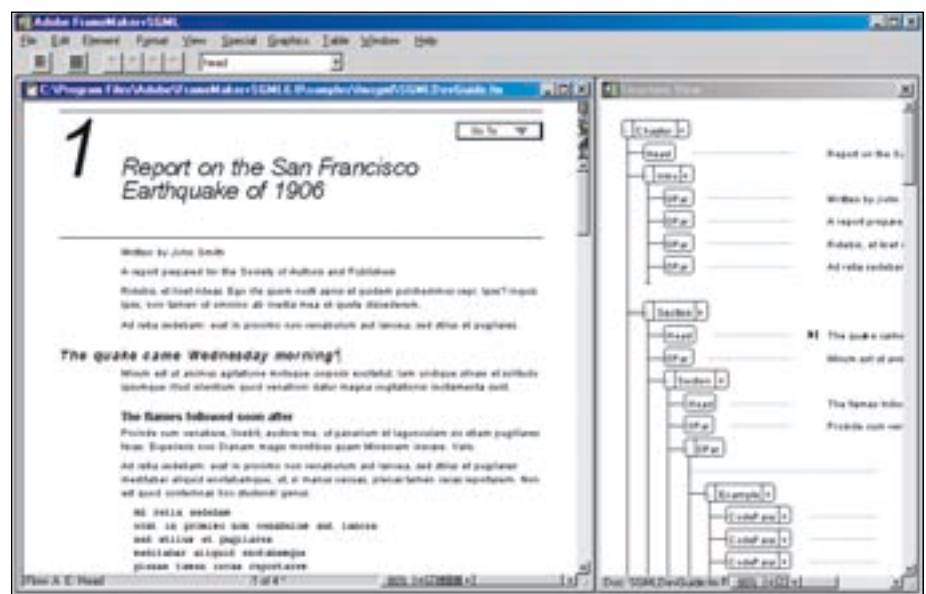
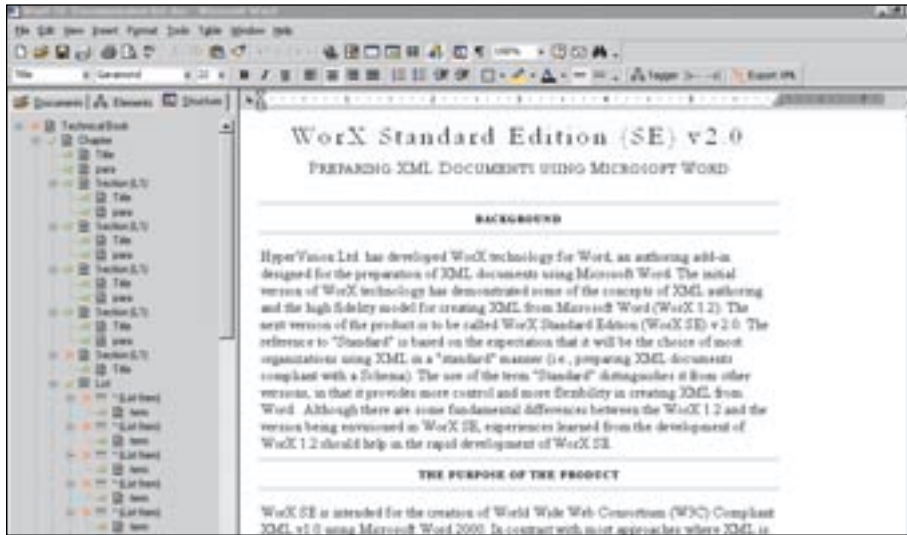


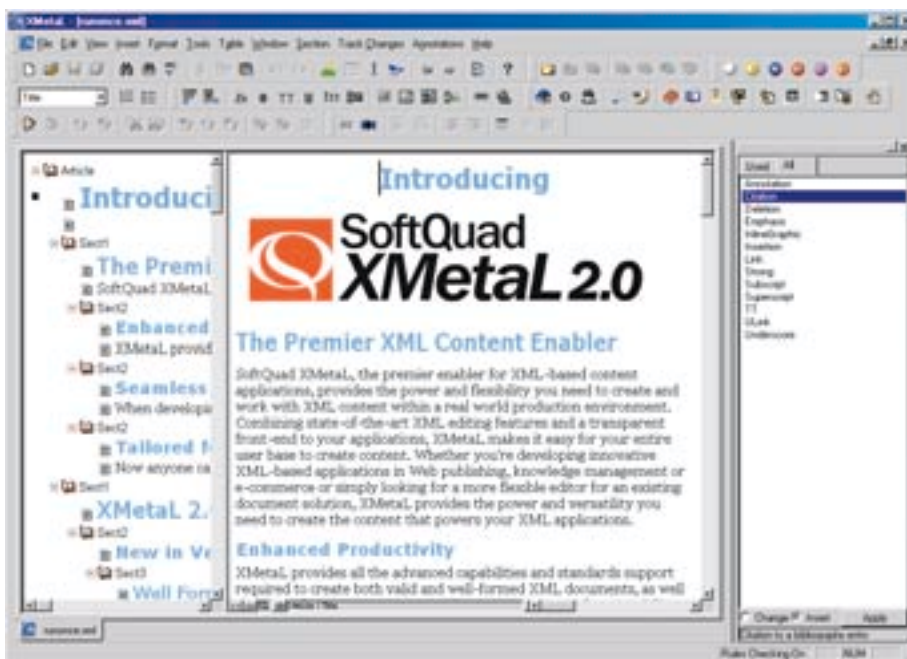
Figure 2. WorX SE's XML authoring interface. The current document's XML structure is represented by the structure tab window on the left.



Such devices can be much more effective when they can both read the words that make up a document and use the structure of the document to present it to the audience. Imagine if your only access to a daily newspaper were through a device that reads the paper word-for-word from page one to the end of the paper. Imagine if the device read every word in the same tone—with nothing to distin-

guish between headlines, body text, and figure captions. This device would be much more useful if you could, for example, direct it to read only the headlines and specify which articles it should read. The device might also vary tone or inflection to identify different parts of the newspaper. Such features are possible if the source format of the newspaper copy is XML.

Figure 3. XMetaL's XML authoring interface. Like WorX SE's authoring interface (Figure 2), XMetaL shows the current document and a structure view window.



Tools for Authors

To take advantage of these new publishing options, authors will need to create XML versions of their content. What changes can you expect when you migrate to an XML authoring tool? If you continue to use print as your main publishing medium, perhaps not much will change. Authoring tools for print publishing will always be expected to provide page layout and formatting support, including running headers and footers, cross-references, and index support. Recently, several tools have appeared that allow you to create XML documents within a traditional WYSIWYG interface. These tools include Arbortext *Epic*, Adobe *FrameMaker+ SGML* (Figure 1), and Hypervision *WorX Standard Edition (SE)* (Figure 2). (Please see Table 1 for more information on software mentioned in this article.)

But as we publish information to various output devices, authors won't necessarily know how their information should be formatted. Authors are likely to become more concerned with content and less with format, as time goes on. As a technical communicator, you may find yourself responsible only for the content of your technical documentation. The format of the information, and even the choice of information that gets published, will depend on the device to which you are publishing and on your audience. The format of the information display for each device is likely to be designed separately from the information authoring.

If you are primarily responsible for content rather than format, you may use an XML authoring tool that does not provide the capabilities of a print publishing program. Such tools, like SoftQuad's *XMetaL* (Figure 3) or Extensibility's *XML Instance*, provide an interface that allows you to create XML documents as well as display how your XML document might be formatted for online viewing.

Catching-Up Specifications

Although the XML specification was finalized in February 1998, several

related specifications for working with XML documents were finalized much later. In fact, several important XML-related specifications are still under development.

Software developers are typically reluctant to support a specification until it is finalized—with good reason! If an application supports an early ver-

sion of a specification that changes during the approval process, the application's XML support may become obsolete. Because of the lead time required to develop software, you can typically expect support for a specification in commercial applications several months to a year after the specification is finalized.

Table 1. Current XML authoring and publishing tools

Tool	Notes	Manufacturer	Web Site
<i>Epic</i>	Full-featured technical publishing package; can maintain documentation source files in XML	Arbortext	www.arbortext.com
<i>FrameMaker</i> , <i>FrameMaker+SGML</i>	Full-featured technical publishing package; XML export capability only (full XML export capability with <i>FrameMaker+SGML</i> , limited XML export capability with <i>FrameMaker</i>)	Adobe	www.adobe.com
<i>WorX SE</i>	Based on Microsoft <i>Word</i> ; combines XML authoring with the user interface and publishing capability of Microsoft <i>Word</i>	HyperVision	www.hvtd.com
<i>XMetal</i>	Not a full-featured print publishing tool; primarily for authoring XML documents; edits XML files directly	SoftQuad	www.xmetal.com
<i>XML Instance</i>	Not a full-featured print publishing tool; primarily for authoring XML documents; edits XML files directly	Extensibility	www.extensibility.com

Table 2. Status of XML and related specifications

Specification	Current Status (see sidebar for explanation)	Status Date	Description
XML (Extensible Markup Language)	Recommendation	February 10, 1998; reissued with minor errors corrected October 6, 2000	Simplified version of SGML (Standard Generalized Markup Language)
XSL (Extensible Stylesheet Language)	Working Draft	October 18, 2000	Language for specifying the format of an XML document
XSLT (XSL Transformations)	Recommendation	November 16, 1999	Language for transforming XML documents to other XML vocabularies or to HTML
XPath (XML Path Language)	Recommendation	November 16, 1999	Language for selecting and manipulating components of an XML document
XLink (XML Linking Language)	Candidate Recommendation	July 3, 2000	Language for creating XML links
XPointer (XML Pointer Language)	Candidate Recommendation	June 7, 2000	Language for linking to specific sections of external XML documents
XML Schema	Candidate Recommendation	October 24, 2000	Language for specifying the structure of XML documents; replacement for the XML Document Type Definition (DTD)

Table 2 lists relevant XML-related specifications and their current status. See the sidebar for information about the World Wide Web Consortium (W3C), the organization responsible for HTML, XML, and other Internet-related standards and the standards process.

What's Next for Authors?

As all the XML-related specifications are finalized, you can expect to see a greater variety of XML authoring tools, as well as more XML-related functionality within those tools. You can also expect XML authoring tools to change the way you work.

Better tools for designing XML documents

The XML Document Type Definition (DTD) is the current mechanism for specifying the structure of an XML document. An XML document is *valid* if it conforms to the rules specified by a DTD.

But DTDs have several shortcomings. The DTD itself is not an XML document, so it is not possible to develop a DTD with an XML authoring tool. DTDs are also too general for some applications. For example, you cannot specify in a DTD that an XML element must be a number. After the XML Schema standard is finalized, better tools should be available to specify the structure of your organization's XML documents, leading to better templates for authors of XML documents.

Better support for print publishing

One of the most anticipated XML-related specifications is the Extensible Stylesheet Language (XSL), an XML-based language for specifying the format of an XML document on a printed page or other display device. After this specification is finalized, publishing software can specify the format of an XML file through an accompanying XSL file.

This could be the beginning of the end of proprietary binary file formats. If you want to change publishing tools, you can take your XML files and XSL files to another tool that supports XML and XSL and continue working.

Better linking support

HTML has accustomed us to creating simple, one-directional links within a document or to other documents. XML expands on HTML linking behavior in several ways, including the following:

- You can maintain the links for a document *outside* the document. Likewise, a

single link can invoke a *set* of links that might be useful to the reader.

- You can specify details about a link's behavior. Should the linked document open automatically, or must the user click on the link to open it? Should the linked document open in the current window (replacing the current document) or in a new window?

- You can link from an XML document to a specific section of another XML document without adding a named anchor (like the HTML <a> tag) to the external document. This has a simple, yet profound, implication. An XML document can link to a section of another XML document *without changing the target document*. No longer do you need to depend on (or request) the author of a document you link with to insert an anchor tag in the appropriate location.

About the World Wide Web Consortium

The World Wide Web Consortium (W3C) is the primary industry standards body for Internet-related technologies. Perhaps the best-known W3C standard is Hypertext Markup Language, or HTML. The W3C also oversees the development of several other Internet-related standards, including Cascading Style Sheets (CSS), standards for Web accessibility by people with disabilities, and, of course, XML.

The W3C standards process is driven by member organizations. A standard goes through three phases on its way to the fourth and final phase, the W3C Recommendation. Only members can provide input to a standard while it is being developed, but the result of each phase is published on the W3C Web site and can be viewed and used by anybody without restriction.

The W3C Standards Process

Working Draft—the first formal public release of a draft specification. Software developers can implement tools based on a Working Draft. However, a Working Draft should be considered a work in progress and is likely to change before it is finalized as a W3C Recommendation.

Candidate Recommendation—a more stable version than a Working Draft.

Proposed Recommendation—a release for one final, short comment period before a specification is finalized.

Recommendation—equivalent to a final specification and guaranteed not to change. This guarantee allows software companies and independent application developers to build tools and applications based on the Recommendation, without fear that the tools or applications will become obsolete because of changes to the underlying W3C Recommendation. This means that any tools that support the XML Version 1.0 specification will always work with XML Version 1.0 documents.

To check the status of the specifications mentioned in this article, or to find out more about the W3C's other efforts, visit the W3C Web site at www.w3.org.



The XML XLink and XPointer specifications provide these and other capabilities. At the time of this writing, these specifications are nearly final. XML authoring tools should begin to support enhanced linking some time after these specifications become final.

More attention to content, less to format

As more writers work in XML authoring environments and XML documents are published on various output devices (hardcopy, online, database, voice), writers will become more concerned with the content that they create and less concerned with format. Formats are likely to be created (and saved as XSL style sheets) when information is published, instead of when it is authored. When a new output device comes along, you can publish to it by creating a new style sheet.

Conclusion

The demand for XML is being driven by the need to publish more information in more ways than ever before. The coming year will be an exciting one for the XML community, as the last of the XML-related standards are finalized and more and better XML authoring tools become available. XML will certainly change the way you publish, and it will probably change the way you write. **1**

Alan Houser is principal partner and co-founder of Group Wellesley, a Pittsburgh, Pennsylvania-based company that serves the documentation needs of technology-based businesses. Alan can be reached at arh@groupwellesley.com.