

EMEDIA
LIVE!EMEDIA
MAGAZINEDVD
PROONLINE
INC.

CURRENT ISSUE

SUBSCRIPTIONS

EDITOR'S CHOICE

EDITORIAL CALENDAR

INFOLINK

CONTACT

[Browse by Subject](#)

information insider

Why Bother With XML?

Robert J. Boeri
Martin Hensel

EMedia, November 1999
[Copyright © Online Inc.](#)

For anyone involved with electronic publishing these days, it is increasingly hard to ignore the Extensible Markup Language (XML). A plethora of active development surrounds the use of XML in areas such as data exchange, electronic commerce, and applications middleware. XML's ability to express customized information simply and hierarchically has given birth to dozens of vendor initiatives and hundreds of startups. Even multimedia has gotten into the act, with first the Synchronized Multimedia Integration Language (SMIL) and now the proposed Boston version. The former is already supported by Real, who promises support for the latter.

But what about XML for documents? XML was originally intended to be a document standard, developed to surmount HTML's "one-size-fits-all" way of expressing information on the Web. XML promised to separate content from form, and to let Web documents define their own information structures. That is, if your Web information needs unique structures (e.g., "<important_note>" instead of ""), you can create your own tag structure. Even better, not only is XML designed for Web delivery, but as a streamlined version of SGML it does away with the high overhead of SGML. However, if your organization never did get on the SGML bandwagon (most never did), and you want to initiate XML document projects, you may be finding surprising resistance.

In an era of information (and tools) overload, nearly everyone wants to work with what they know and avoid licensing, learning, or supporting another tool. This attitude is common among users of Corel WordPerfect, Microsoft Word, and others. Vendors of Web-CD library delivery products continue to support popular word processors or desktop publishing systems as native input. Hynet's Directive publishing product provides conversion of Microsoft Word and Adobe FrameMaker documents to HTML or XML (and thence to Web-CD delivery). Inso's Dynatext supports both Word and WordPerfect. Enigma's Insight does likewise and even includes both Acrobat PDF and FrameMaker+SGML (enabling still richer PDF structures). If you can achieve slick Web-CD delivery and keep your familiar tools, why bother with XML?

do you want links with that?

If you have been distributing content on optical media or networks using earlier versions of Adobe Acrobat, you were accustomed to handcrafting most links within those documents. With Acrobat 4.0, you get a simple plug-in that creates a multitude of links within Microsoft Word 97 (or PowerPoint 97) documents. Simply fill out a menu of link types, and section headers appear in your table of contents link to sections within the document.

URLs become live links to the Internet, and that is just the beginning. Acrobat PDF files created thusly are not simply pretty "digital paper"--they have an undeniably rich structure. If you can create these rich links from a Microsoft Word document automatically, why bother with XML?

fashion models

Remember the "Document Type Definitions" (DTDs) required in SGML and optional in XML? XML requires a document to have an implicit DTD, and originally that flexibility was expected to speed its use. But just because you can define many markup languages to express the same kind of document content, should you? Should Ford, GM, and Toyota each define their own document model for interchanging information with their suppliers just because they can? No, but getting competitors to agree on document vocabulary isn't easy.

To make things worse, the DTD itself as a document model might be going out of fashion. As data interchange applications seized on XML, they uncovered a secret largely kept within the SGML community before XML: DTDs don't do as good a job as database systems to define rich data types. Range checking of data types (e.g., house prices of type "currency" between \$300,000 and \$500,000) illustrates an inadequacy. One new challenger to DTDs: XML Schemas.

If you can agree on which approach to modeling documents, what tools are there available to build those models? Right now you don't have many choices. One of these is "Near & Far Designer" (<http://www.microstar.com>) a graphical tool originally designed for the SGML world and now with XML support, at a list price of around \$800. Another, "XML Authority," (<http://www.extensibility.com>), has been in 1.1 version since July. It supports both DTDs and all major emerging schema formats, and lists for less than \$100. Still, if you can't get the model right (or at least express it confidently), why not wait before taking the XML plunge?

making documents look right

One of SGML's biggest failures was its inability to define a commercially acceptable standard for applying formatting to documents. In fairness, the Document Style and Semantic Specification Language (DSSSL) failed because describing document look-and-feel is extremely difficult. Since the XML equivalent, the Extensible Stylesheet Language (XSL) was proposed in 1997, XSL has fissioned into descriptor and document transformation formats and Adobe Acrobat 4.0, delivers faithful electronic document renditions, and does this without XML. So, why bother with XML? Stay tuned.

Robert J. Boeri (bboeri@world.std.com) and **Martin Hensel** (mhensel@hensel.com) are columnists for *INFORMATION INSIDER*. Boeri is Information Systems Publishing Consultant at Factory Mutual Engineering of Norwood, Massachusetts. Hensel is president of Texterity Inc., a Newton, Massachusetts-based consulting firm that builds SGML-based editorial and production systems for publishers, corporations, interactive services, and compositors.

Comments? Email us at letters@onlineinc.com.

[\[EMedia Home\]](#) [\[Subscriptions\]](#)

Copyright © 1999, [Online Inc.](#) All rights reserved.
info@onlineinc.com