



## Content Rule #4:

Content without context is just more 0s and 1s.

[Home](#)[Magazine](#)[Conferences](#)[Advertising](#)[About](#)[Site Map](#)

## XML Across the Publishing Lifecycle: Tools & Strategies to Promote Success



**Robert J. Boeri**

*October 2001* | A funny thing happened on the way to the Web. XML, designed as a Web standard to replace HTML, was intended to give every Web page the ability to define its own tags, breaking free of the Procrustean bed of presentation-only markup. Since the release of the XML standard in February 1998, most XML applications have instead been for data interchange and ecommerce.

That may be changing. What barriers must you overcome to succeed in XML content publishing? Let's start with a metaphor. Imagine that you want to build a new XML house. Having heard about the benefits of XML construction, you call up an architect and are told: "Using XML construction has benefits you can't achieve with any other form of construction. You're wise to get an early start. However, there are a few issues you should know about. We can buy the raw materials, but you will of course have to modify them." You reply: "Modify the materials, you mean like make my own nails?"

"Why yes. One of the many advantages of XML is that you can customize it. In fact, you must do so. But don't worry, there are lots of tools available to help with this customization."

If this is how they built houses, we'd all be living in tents. Setting up a realistic XML publishing system isn't simple, but today's standards and vendor offerings are making things easier. Although still more complicated than a trip to Warehouse Building Supplies, you no longer have to make your own nails. XML tools, standards, and vocabularies are all maturing, and publishers are making broad commitments not only to XML, but to content management system infrastructures for streamlining the entire publishing lifecycle.

Think of this lifecycle as starting with document model analysis and design, onward toward authoring and editing, then assembling content chunks to publish products. The cycle is then repeated as the designs evolve and the content is revised— with this caveat: Nearly everyone's content is "legacy," in a format other than XML. Converting that content to XML is not trivial. Moreover, the lifecycle is so complex that most organizations choose some automated system to manage it.

### WHAT'S THE BIG DEAL?

Let's begin by asking, "Why change your present system at all?" You could be noticing XML on competitive Web sites or in RFPs. Perhaps you notice it's getting harder to support your content's proprietary formats, or you can't derive products you need from them. Maybe the process of managing content through your whole lifecycle has become a hassle that only a few of your top talent understands. These warning signs suggest it might be time to upgrade your content to XML. After you've succeeded, these benefits can accrue:

- Reduce your dependence on proprietary formats
- Create many products from single-source XML assemblies
- Achieve early strategic competitive advantage by implementing upcoming standards, tools, and technologies

- Start working more closely with your customers and suppliers who already use XML

XML may not be for you. Proprietary authoring tools like Word and FrameMaker can provide limited cross-media outputs. If those suffice, perhaps you can avoid XML or postpone the move. However, if you decide to implement XML, consider non-technical issues too, such as your corporate culture. Does it resist change, or is there a grass-roots desire to modernize? Do you prefer gradualist approaches—piloting one part of the lifecycle before moving on to the whole? Or do you want to implement comprehensively? In both cases, you'll need to evaluate competing vendors and follow standard vendor selection procedures. You'll also need to decide just how to balance the use of in-house staff with vendor consultants. Consultants achieve quick but expensive re- sults; in-house expertise is available over the long haul.

## ANALYZING, DESIGNING & APPLYING YOUR XML VOCABULARY

There's nothing magical about XML vocabularies, also called document models, DTDs or Schemas. These are simply sets of rules defining the structures found in your content. You can use existing models designed by industry groups (like DocBook for technical information), or you can create your own. Two excellent, inexpensive modeling tools are Altova's XML Spy and Tibco's Turbo XML. Spy is designed for Windows users; Turbo XML is written in multiplatform Java. Both can do much more than just model building; they can help convert content, produce graphical model views, and document the models.

Just as we remodel houses, our XML models evolve too, especially after we apply them to all legacy content. Modeling and converting content is often an iterative process. Even if you hire a consultant to develop your initial model, investing in a modeling tool will help you evolve the model as necessary.

**Tradeoffs:** If your environment includes a mix of Macintosh, UNIX, and Windows systems, Java-based tools may work on all three. This reduces your total cost of ownership, including training. However, these tools are not tuned to any one platform and will run slower than tools designed for just one operating system.

Developing your model and converting your content to XML requires a deft combination of content analysis, strategic thinking, and business planning. Does the structure of your content fall into one or more categories, each covered by the model(s) you developed? Do those categories satisfy your anticipated product requirements? If your content already exists as HTML or can be easily converted to HTML, a freely available tool called HTML Tidy can help convert that to XHTML as a first step to your final conversion. XHTML itself can begin providing immediate benefits as input to ebooks and as a preparation for delivery to wireless devices. If you use Acrobat, Adobe and Texterity are developing tools to convert PDF to XML.

If the prospect of conversion with existing staff makes you uneasy, consider a vendor experienced in both model design and conversion to do the job. Costs start at around \$10 per standard page, depending on your content and your XML model(s). Will you run legacy and new systems in parallel for a while? If so, be prepared to capture all legacy changes and apply them to your XML system.

**Tradeoffs:** Think of Tidy as a free, but 80% solution. Someone will need to review every "tidied" file. The resulting valid XML will still require additional model information.

## AUTHORING AND EDITING XML DOCS

Selecting an XML authoring tool requires a fundamental choice: Do you want a native XML tool like SoftQuad XMetaL or Arbortext Epic that works with XML directly, or do you prefer hybrid tools that add XML awareness to a word processor or publishing tool your organization already uses?

Examples of hybrid tools include BroadVision's One-to-One Publishing system with its customized version of Microsoft Word, Adobe's FrameMaker+ SGML, WorX SE, and S4/TEXT. Although using Word to create XML is appealing, it's interesting to note that

Microsoft Corporation itself purchased a site-license to XMetaL for its in-house XML authoring needs. The more a tool looks and feels like a word processor, the more it is likely to depend on word processor styles. Styles, however, are notoriously free form, and ensuring their strict use to generate XML is challenging. Word, for example, will let you insert styles virtually anywhere in a document; XML is very strict about which elements are valid in each context. Thus, getting from a loosely styled word processing document to a validly structured XML document is not an easy achievement. If you insist on an MS Word solution, consider eXtypes, a tool designed to help editorial and production personnel clean up MS Word files so they stand a better chance of yielding valid XML.

One choice for having your word processor and getting valid XML, too, might be Corel WordPerfect. Now available as WordPerfect Office 2002 Professional Edition, WordPerfect has nearly 10 years' experience supporting XML (and SGML) natively. As with hybrid tools, some setup will be required, but the result might combine the best of word processing and XML.

**Tradeoffs:** Hybrid XML authoring tools are initially more familiar than native tools, so selecting them may reduce (not eliminate) training. However, be very wary about support costs. Ask for customer references. Also ask for proof that your legacy word processing content can easily be imported or converted to XML based on their use of styles. Lastly, remember that every time your DTD changes you'll need programming assistance to upgrade these hybrid tools and possibly your content, too.

## ASSEMBLING & PRODUCING PRODUCTS

Unless you can transform and render your XML content as profitable products, XML remains an academic exercise. The XSL family of standards is key for XML styling and transformation. Most XML delivery systems use XSL, and some even include rudimentary XSL development tools. Writing XSL is akin to learning JavaScript in-depth; unlike JavaScript though, you are unlikely to find libraries of XSL style sheets to reuse since your XML models and content are unique. eXcelon Stylus Studio provides graphical tools for creating XSL style sheets within a rich graphical environment, and this inexpensive suite can speed learning and development.

Arbortext Epic also goes beyond authoring to multiplatform delivery. Enigma's 3C platform provides a complete strategic framework for ecommerce publishing to Web and CD. Enigma makes robust use of XML, but it can also work with outputs from Word and FrameMaker.

**Tradeoffs:** Although the XSL standards family is powerful and key to cross-media publishing, few delivery tools support these standards fully today. Still, any investment you make to learn XSL will pay dividends.

## MANAGING THE XML INFRASTRUCTURE

Any lifecycle XML system will be complex. You'll want to manage your models, content, and metadata (information about your content). If you're already using a file versioning system like Visual SourceSafe, you could continue with it for a while. That inexpensive choice will not, however, provide workflow management, content assembly, validity checking, or enhanced searching. Robust XML-aware content management systems will provide those services. Systems available from Documentum, BroadVision, XyEnterprise and others will become appealing as your XML content grows. Some are integrated solutions; others allow (and require) you to select components and integrate them into the system. All will need some consulting services.

**Tradeoffs:** Native content management systems take advantage of XML and can save many versions of documents with little extra space (they often save only changed elements). Such systems also provide more focused element searching. However, native content management systems are more expensive and more difficult to set up than simpler file-versioning systems. Like building a house, establishing an XML publishing system requires planning and discipline. The results can give you shelter from the elements of proprietary systems.

---

## 10 Considerations Before Making the Move to XML

1. Why change from your present proprietary systems to XML?
  2. How will you switch over from a pilot to the full system?
  3. How will you convert your legacy content to XML?
  4. What is your organization's tolerance for change?
  5. Have you begun developing a metadata strategy?
  6. How consistent is the structure in your legacy content?
  7. Do you prefer a native or hybrid XML authoring solution?
  8. Can you use an industry-standard XML vocabulary, or must you build your own?
  9. How do you want to balance consulting services with developing in-house expertise?
  10. What kinds of new products and delivery systems do you anticipate developing?
- 

## 10 Things To Ask XML Vendors

1. Is your product part of an integrated content management solution, or must I mix, match, and integrate applications?
  2. What solutions do you have for each of the stages of my workflow and lifecycle?
  3. Does your product let me begin small and then grow to a total solution?
  4. How easy is it to develop and maintain a custom workflow?
  5. How do you license the product? Must I buy named seats or simply the quantity I'll use at any time?
  6. Do you sponsor user groups or online discussions for users of your products?
  7. Can you give me customer references whose needs and documents are similar to mine?
  8. Can you identify opportunities to integrate with corporate databases?
  9. Does your product integrate with mainstream applications like email?
  10. Does your system search and index XML elements and attributes?
- 

## Aspen Makes the XML Leap

Are any firms with a history of print and legacy files moving to XML successfully? I asked Cimarron Buser, Marketing VP at Texterity, a provider of outsourced e-publishing services in central Massachusetts, if he could name any successful efforts. He named Aspen/Loislaw, a division of Wolters Kluwer, with whom Texterity works. Aspen, headquartered in New York City, is a U.S. provider of primary and secondary source material for legal research in print and delivered by subscription over the Internet. Buser said this legal publisher has a decades-long tradition of print publishing, yet made the transition profitably from print-only to XML. Buser said that every book seems to have its own procedures deeply embedded in its editorial and production culture. Aspen managed to maintain that culture, including a high esteem for print, while moving to single-source XML publishing. Buser insists that XML publishing isn't for everyone. No one should make the move unless their documents have a disciplined structure (or at least consistent formatting) and their plans include deriving several products from a single authoritative source.

Buser cited Aspen/Loislaw publisher Marc Jennings' assessment of the pros and cons of this move. The benefits include accelerated product development cycles, the ability to produce more products from a single, vendor-neutral source, and streamlined production cost savings. None of this comes without cost, however. Jennings also said that the startup costs—especially the initial one-time data conversion to XML—had to be weighed, along with the need to train and then retain skilled employees.

Not every successful XML project needs to purchase every bell and whistle either. Instead of buying an in-house content management system that would have been costly and required even more in-house expertise, Aspen uses Texterity's BookBank service. This is essentially

an Application Service Provider model whereby Texterity provides Web-based access to Texterity's file management system for editing, XML parsing to assure valid XML throughout the lifecycle, and PDF generation to provide print-quality renditions of the XML source. "When you consider that Aspen had only a few IT employees and was able to make this move, anyone can succeed provided they analyze costs and benefits," said Buser.

**ROBERT J. BOERI** ([bboeri@ieee.org](mailto:bboeri@ieee.org)) is an information architect at a Boston-area financial publishing company.

Comments? Email letters to the editor to [ecletters@onlineinc.com](mailto:ecletters@onlineinc.com).

---

Copyright 2000-2001 Online, Inc.  
213 Danbury Road, Wilton, Connecticut 06897-4007  
203/761-1466, 800/248-8466  
Fax 203/761-1444  
[info@onlineinc.com](mailto:info@onlineinc.com)