

Casting a critical eye on the Next Big Thing in technical publishing.

The ABCs of XML

By SARAH O'KEEFE, Associate Fellow

To succeed as a technical communicator, you need writing ability, domain expertise in your product area, and a reasonable understanding of publishing tools and technologies. If you want to remain relevant (and employed), XML is rapidly becoming part of the required knowledge set. You don't necessarily need to become an XML expert, but you do need to understand what an XML workflow looks like and whether it makes sense for your organization. This article discusses the three most important reasons that you should consider XML.

Automation

In traditional authoring tools, we write content and apply formatting to that content as we go. In an XML-based workflow, we create content only. Formatting information is added later when the content is rendered into its final output format or formats. This results in the following:

- Information is more flexible because it does not have formatting attached to it. When you are ready to publish, you select the output type you want and generate the output.
- Writers lose the ability to tweak formatting as they are writing. Although

many writers find this new reality a bit painful at first, it does increase the writer's productivity by eliminating formatting responsibilities.

- The automated formatting can be attractive, but it's not as good as output produced by hand-crafting each deliverable.
- The publishing process is automated and very, very fast.
- Automating the publishing process lowers the



cost and effort of creating content, especially for organizations that produce content in multiple languages. In an industry survey conducted by my company, 70 percent of respondents reported that implementing structured authoring reduced their cost or effort of developing content (O'Keefe & Pringle 2009).

Baseline Architecture

There's nothing new about the idea that we create content once and publish it into multiple formats. Single-sourcing workflows have been around for many years. However, many of the older workflows start with proprietary file formats, such as Word or FrameMaker, and then transform them into other formats, such as HTML. If you choose XML as the baseline architecture for your content, you are storing information in an open format that does not belong to any specific



vendor. By doing so, you achieve the following:

- You can create content in a variety of XML editing tools.
- You can manage content versions and workflow in a variety of content management systems.
- You can publish content through a variety of publishing tools and technologies.

You reduce or eliminate what application developers call *stickiness*. Software is sticky when it is difficult for a software user to move from one application to another. Stickiness is generally bad, unless you are in the business of selling software.

Most software developers are now familiar with XML; they may use it for configuration files or other parts of their application development process. Storing content in XML may make it easier to get content contributions from software developers.

Some of the information that you work with may need to be available in several different places. For example, the error messages that appear in software might also need to be in an error message reference. It's possible to create an XML file that becomes the source file both for the software error message and for the error message appendix. Similarly, the technical support organization might have information in their knowledge base that should also be included in product documentation. Instead of burying information in a proprietary tool, you can use XML to expose content to many different uses.

I expect XML content to play a critical role as technical communicators integrate their professional information with user-generated content. If your organization provides user forums, a wiki, and other venues for the user community to

“The technical side of XML is not much more difficult than HTML.”

discuss your products, you need a way to combine that information with the official, corporate documentation. XML will provide the bed-rock for these efforts because of two critical features:

- You can embed metadata in XML files and use the metadata to determine what information is displayed to end users.
- You can control the presentation of XML content so that you can clearly separate official information from user-generated content.

Consistency

Document consistency is an important component of document quality. All documents from a single organization should be consistent in:

- Terminology—Industry terms should be used consistently throughout all of the content.
- Look and feel for each document class—For example, all white papers should have the same design, all web pages should have the design, and so on.
- Organization—That is, if the style guide indicates that warnings should precede the step that the warning applies to, the warnings should always precede the step. If a procedure has prerequisites, those prerequisites should always appear in the same place.

Document consistency makes readers feel more confident that the information is correct. XML is especially useful for ensuring consistency of look and feel and organization. Consistency of look and feel is achieved by creating

document design standards and then implementing them in the automated publishing workflow—documents are always generated the same way and have the specified look and feel.

When you author in XML, you have a set of rules that govern the structure of the information you are creating. For example, the rules can specify that:

- A heading must always have at least one paragraph that follows (no stacked headings).
- A bulleted list must contain at least two items (no single bullet lists).
- A lesson plan must have at least two objectives.

The exact rules will depend on your content and your stylistic standards. As you create content, you can verify that your information follows the rules that you have established; this process is called *validation*. One basic principle of XML is that you cannot process files that are invalid. Instead of asking an editor or a peer to review content to ensure it conforms to the required structure, the authoring software can prompt you to insert the needed components at the appropriate location. This is called structured authoring—a publishing workflow in which consistent organization of information is defined and *automatically enforced*.

Do You Need XML?

Your organization may or may not need XML right away, but you do. Our industry survey found that nearly 30 percent of respondents

are already using structured authoring; by 2010, a majority of respondents expect to be using XML. You need to learn and understand XML so that you can keep your career options open.

The technical side of XML is not much more difficult than HTML; if you can handle a few HTML angle brackets (<p> or <h1>), you can learn XML. The concept of structured authoring is a bit more challenging. If you have worked in an environment where the template was rigorously enforced, it's only a small step over to structured authoring. If, however, you don't like using styles and prefer to format everything as you go, you are going to *loathe* structured authoring.

The question of XML is more complex for companies. Implementing an XML-based publishing environment is difficult and usually expensive. It requires a new set of tools, new skills, and a new mindset from the content creators. Implementation generally takes 6–12 months and often requires outside consultants. Given those barriers, there needs to be a compelling rationale for making the change. That said, organizations should not wait too long before deciding on an XML strategy. In addition to significant cost savings, XML provides the foundation on which you can build more advanced content platforms. For example, you could:

- Integrate content across departments.
- Integrate with user-

generated content.

- Integrate with software products.
- Personalize content based on user profiles.

In other words, the ABCs of XML are just the beginning of the road. **❶**

REFERENCES

O'Keefe, Sarah, and Alan Pringle. 2009. *The State of Structured Authoring in Technical Communication*. Scriptorium Publishing. PDF available for download at www.scriptorium.com.

Sarah O'Keefe (xmlstrategist@scriptorium.com) is founder and president of Scriptorium Publishing Services Inc. (www.scriptorium.com) based in Research Triangle Park, North Carolina. The company is focused on implementing tools and processes to optimize publishing workflows. Services include developing and deploying XML-based structured authoring environments, configuring authoring and publishing tools, and providing technical training. Her publications include Publishing Fundamentals: FrameMaker 7, The WebWorks Publisher Cookbook, Technical Writing 101, FrameMaker for Dummies, and numerous white papers (available at www.scriptorium.com/papers.html). Sarah is an STC Associate Fellow and member of STC's Carolina, India, TransAlpine, and UK Chapters and of the Consulting and Independent Contracting, Europe, and Management SIGs.