

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

## What Do Movable Type and XML Have in Common?

BY SARAH O'KEEFE, Associate Fellow

In the mid-1400s, Johannes Gutenberg developed a printing press that used movable type. He began printing the famous Gutenberg Bible around 1452. Movable type was the enabling technology that made printing economically feasible.



Instead of creating printing plates as single-use, page-based stamps, Gutenberg broke down the granularity of printing to the individual letter and made the letters themselves reusable. This technique greatly reduced the cost of printing—before movable type, books were precious, rare objects and were usually copied by hand over a period of months or years. A large library, such as one owned by a prince, a wealthy monastery, or a leading university, might boast of owning 1,000 books. Today, many of us have a comparable number of books on our shelves at home.

Gutenberg changed the economics of information distribution, and I imagine that many of the artisans who devoted their lives to creating glorious, hand-copied books complained bitterly about boring printed books. It's also worth noting that many of Gutenberg's Bibles were illuminated (by hand) after printing, and those illuminations were

specifically commissioned by the buyer. That is, there was a transitional period in which printing and illumination were combined to create the finished book.

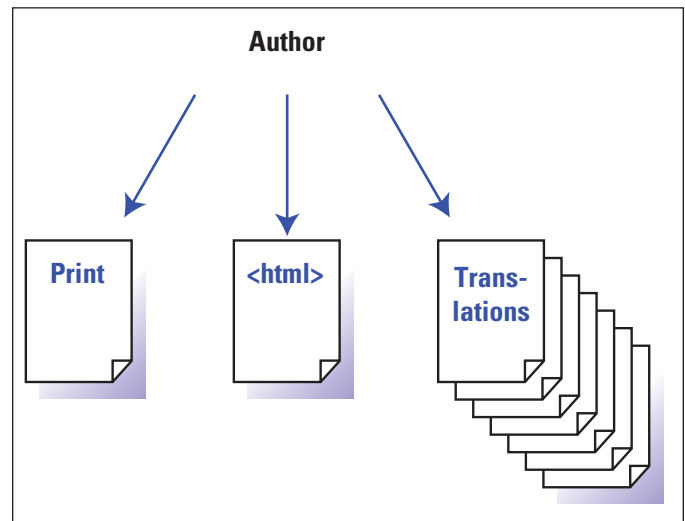
Like movable type, XML changes the economics of publishing. But where movable type changed the economics of a mechanical process—creating printed copies—XML changes the economics of content authoring, formatting, and customization.



### The Economics of Authoring with XML

Since the 1980s and the adoption of desktop publishing as the dominant workflow, authors have been responsible for not just writing and editing but also document formatting and production—the process of finalizing content for delivery. With XML, production

Figure 1. In a traditional workflow, each output format in each language requires a production effort.



responsibilities are stripped away, and authors focus solely on writing and editing their content. XML also enables the creation of *enforceable* document templates. The decrease in authoring responsibilities results in increased efficiency and productivity for authors.

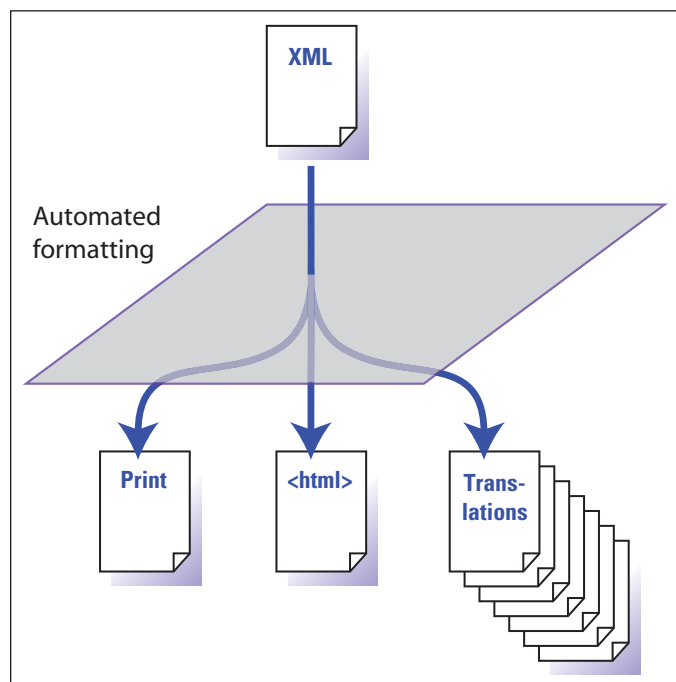
It's worth noting that we are talking here about authors who are equally skilled in the two workflows. A person who is familiar with desktop publishing and just learning about XML-based authoring will face a significant learning curve before becoming productive in XML authoring. Perhaps it would be better to say that XML-based authoring *has the potential* to increase efficiency and productivity over what's possible in a desktop-publishing environment.

### The Economics of Formatting with XML

In an XML-based workflow, content is formatted automatically when the final output is produced. Instead of formatting content as it's written, formatting is added as a separate layer. From a cost perspective, this results in eliminating an ongoing cost (formatting while authoring), and replacing it with a one-time cost (development of automated formatting). The cost savings realized from automated formatting are multiplied when you have multiple output formats and, especially, multiple output languages.

For very small organizations with no localization requirements, the cost of implementing XML may exceed the savings realized today. Over time, the

**Figure 2. In an XML workflow, authors create XML files without formatting instruction, and production is automated.**



tools and technologies will improve and the implementation cost will drop, making XML appealing to smaller and smaller groups.

For organizations with large amounts of content, however, the cost savings are compelling. Manual page layout and copy-fitting may result in higher-quality printed documents but, like medieval manuscript illumination, it is destined for obsolescence because it costs too much.

### The Economics of Customized Content with XML

Content customization is a thorny problem in traditional workflows. The best possible solution has been to use conditional text or to build tags to flag information as conditional, and then to generate several versions of the output files.

With XML, it's possible to ship a single set of content and then display different information depending on the requestor's profile.

You can achieve a limited amount of customization without XML. If you need a dozen or fewer variations of the final output, the conditional text/build tag approach will likely work. XML, however, gives you the ability to do the following efficiently:

- Set up large numbers of variations.
- Filter the content when the user requests information, rather than when the content is published.
- Create multidimensional conditional content, such as information that is filtered by customer and also by platform.

It's not impossible to do any of the preceding items without XML, but it's prohibitively expensive.


### The Vision of Content Integration

Automated formatting and more sophisticated conditional processing are important but incremental steps forward in publishing. XML technology can also provide the foundation for a new approach to technical content. The automated processing enabled by XML is what makes automated formatting and production possible. But we can take XML content much farther and begin to use it as the foundation for many different types of uses. For instance, imagine an XML file that contains error messages, descriptions, and troubleshooting strategies. This content would be processed to create an error message appendix in documentation, but it would also be used to generate the error messages displayed by the software. That is, the XML file would be the source file for technical content and be an integral part of the software.

It's also possible to integrate user-generated content, such as wikis, blogs, and forums, with the documentation created by professional technical writers. Here, XML can serve as the enabling technology that makes it possible to mix content from many different sources and deliver it to the end user in a unified presentation. Provided that each type of content is stored as XML, we can build software that reaches into the various types of content and extracts the relevant chunks of information. This

content integration is the new frontier for XML-based information delivery, and I look forward to seeing it evolve over the next few years.

### Opening Up New Possibilities

More than 500 years after the fact, it's easy to see the impact of Gutenberg's invention. From our vantage point inside the upheaval caused by XML, it's much harder to assess where XML might take us. I believe, however, that we have barely begun to explore what structured content can do for us. 

### SUGGESTED READING

For an excellent description of Gutenberg's impact on printing and the historical context, refer to Jean-Marie Dodu's *The Gutenberg Bible: A Commentary* (Paris: Editions Les Incunables, 1985).

The British Library has a Gutenberg Bible online, along with background information: [www.bl.uk/treasures/gutenberg/homepage.html](http://www.bl.uk/treasures/gutenberg/homepage.html).

---

*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. She is senior member of the Carolina chapter and of the Consulting and Independent Contracting and Management communities.*

*Thanks to Terry Smith of Scriptorium Publishing who created the graphics for this article.*