

Why go to the trouble of converting the document from PDF to XML if it's ultimately going to be presented on the Web site as a PDF document? The advantage of converting the document to XML can be seen in Figure 5. When the final document is an XML document, you can present it as PDF, HTML, RTF and a variety of other formats including mobile and voice displays. In our case, that is a major advantage.

Our Web site contains approximately 50 documents ranging from 50 to 100 pages. Most of these are in a PDF final format and not in HTML at all. Just to produce these same documents in an HTML format would nearly double the size of our Web site in files and pages. Plus, it would double the complexity of maintenance since we would have at least two final versions (one in PDF and one in HTML). In many cases, we would have three final versions (a Word document as well) and frequently a PageMaker® or FrameMaker® version for printed publications.

Looking back, Figure 4 illustrates the impact of these multiple versions on workflow. For example, every time the content changes in the Word document, a web developer has to edit every HTML page impacted by the change and a new PDF has to be produced. Something as simple as changing the title of the guide could involve changes to 50 different HTML files and manual proofreading of documents (Word, HTML, PDF) to verify consistency.

The complexity increases when browser incompatibilities are considered. Take a simple example of providing a text-only version of the document for users who do not use graphical browsers. Not everyone uses Internet Explorer, Netscape, Opera or Mozilla; some people use Lynx or other non-graphical browsers. Now we need 50 HTML pages for the graphical version, and another 50 for the text-only version. Furthermore, if we decide to use browser-specific features (IE6 and Netscape 4 for example), we have again increased the number of HTML pages on our site.

XML alleviates these problems by:

- Creating a single source for the final version of your content (it's in one XML file as shown in Figure 5)
- Using multiple XSL files to selectively transform the XML document into the appropriate output format (HTML for graphic and text-only browsers, PDF, etc.)

Rather than having multiple source files and hundreds of HTML files, a document like the 50-page **Gateways Guide** would have one source file and perhaps 5-10 XSL files to produce the multiple output formats. As content changes occur, the change is made in the single XML file and then automatically and immediately propagated to the various output formats. A Web developer does not have to modify dozens or hundreds of HTML pages; no one has to generate a new PDF using Adobe® Acrobat®; and no one has to proofread all the various formats to ensure the change was applied consistently.

Cocoon offers additional features to further ease these maintenance issues. For example, one-line parameters identify different browsers and automatically direct users to different stylesheets appropriate for them. Instead of 50 different HTML pages for each type of browser, you have one line and one additional XSL file. (The next section examines these Cocoon features and XSL files more closely.)

Unlike the HTML-based architecture which suffers from increased maintenance impacts as the Web site grows in size and complexity, an XML-based architecture actually levels off. There is a limited impact on maintenance as the site gets bigger and more complex.

The ease in managing the content comes from the basic property of XML that provides a total separation of content (source) and style (output). The content or data resides in a single XML document and different XSL stylesheets present that data differently.