



Working with XML

This column focuses on XML and its use in the technical communication field. It explores the benefits and promises of XML, and focuses on how to get XML to deliver on these promises. Please send your comments, questions, and suggestions to france.baril@ixiasoft.com.

BY FRANCE BARIL, Column Editor

Information Models and XML

The XML overview printed in this column in June 2006 was meant to establish the base for further exploration. Now I want to explore the defining feature of XML: the ability to specify your own tags. In less technical terms, you get to define your very own specific information model.

Know Your Content

If you have been in this industry for a while, you know that being organized, at least when it comes to content, is of utmost importance. Whether you work as a contractor or employee, in a small company or large organization, you probably already have a system for organizing content. This system may be implicit or explicit, depending on your work environment, but you know where you're going.

There are different ways to organize content in a deliverable, and your information model probably reflects how users access information as well as how it is created. Some people work with simple information models; they think in terms of chapters and sections. Others need to organize information in more detail; they may speak in terms of functionalities, user tasks, or system tasks. Some are even more specific; for example, they might have rules that forbid writers to include more than twelve steps in a task.

Systematizing and reinforcing your information model is a good way to make sure that you and your team stick to the rules. Doing so improves consistency, tends to enhance usability, and facilitates collaboration with other authors without creating a document that looks like it was written by five people.

XML supports and reinforces your information model by using document

XML, by reinforcing the content structure, facilitates reuse.

type definition (DTD) or XML schemas.

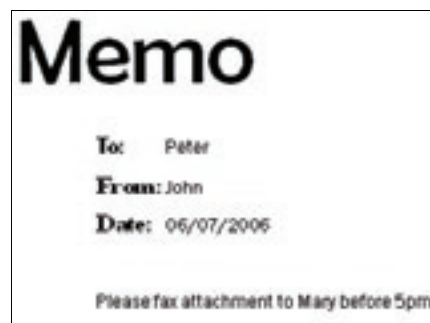
If you can write down the structure or rules that you always follow, or wish you followed, you might be ready to take one of the first steps toward moving to XML: defining a DTD or XML schema, or choosing an existing one that meets your criteria. While you're at it, you might also want to update your current structure and rules in order to profit from the new opportunities XML offers. Let's take a look at some of the themes that you might want to explore further.

DTDs and Schemas—An Overview

Choosing and defining your DTD/XML schema is the technical equivalent of determining your information model.

DTD/XML schemas define what type of information may appear in the content, where it will appear, and in what order.

Let's use this memo as an example:



A DTD or XML schema could define the structure of this document in this way:

Memo contains

- *To*, followed by
- *From* (optional), followed by
- *Date* (optional), followed by
- *Message*

Please note that I am paraphrasing the DTD/XML schema languages for more clarity. To learn about DTD and XML schema syntax, see www.w3schools.com.

The XML could look like this:

```
<memo xml:lang="en-ca">
  <to>Peter</to>
  <from>John</from>
  <date>06/07/2006</date>
  <message>Please fax attachment to
  Mary before 5pm.</message>
</memo>
```

Differences between DTD and XML Schemas

XML schemas are successors of DTD schemas. XML has some advantages over DTD that lead many to believe it might end up replacing DTD altogether. Some advantages of using XML schemas over DTD are that XML schemas are written in XML and are therefore easier for computer programs to manipulate, make it easier to describe permissible document content, allow for more levels of data verification, and make it easier to work with data from relational databases.

However, DTD schemas have been around longer and are still better supported by many tools, are easier for non-technicians to learn, and are easier for human beings to read.

What to Look for in an XML Information Model

Your XML information model should contain the basic document structure and important semantics information, such as vocabularies used in your industry.

The basic document structure represents your content philosophy, your content template. What defines your document structure? Here are a few questions you could ask yourself:

What are the basic building blocks in my documents and how many do I have?

Examples of building blocks include procedures, descriptions, API (Application Programming Interface) methods, training overviews, chapter summaries, and illustrated parts.

Also look at the similarities among the structures of these blocks. Should the blocks be defined separately? Are some so similar in structure and semantics that they could be grouped? For example, are functionality descriptions and concept definitions both made of random sections and paragraphs, or does one have a more specific structure (e.g., a definition followed by a detailed description, followed by an example)? If both have similar structures and behaviors, you might decide to define them with the same XML elements.

Are you in an industry that already uses a specific standardized DTD schema?

The choice of your basic document structure should be based on multiple criteria, including the way you structure information, the standards that other people you work with or others in your industry already use, and the goals that you want to reach.

Following industry standards increases usability since users get accustomed to reading similarly structured documents; it also makes sharing content with business partners easier.

What content can be reused?

Is there content that you type over and over again? Is there content that you reuse systematically? Is there content that is so similar that you wish you could reuse it?

XML, by reinforcing the content structure, facilitates reuse; it helps to ensure that all the necessary information is included and makes it easier to find this information when it is needed.

There are different types of reuse. I won't try to list them all here, but will mention some opportunities worth looking into.

Reusing the Document Building Blocks

By defining a modular content model where modules (sometimes referred to as *topics* or *sections*) are your basic units for reuse, you might enhance your opportunities for reusing them in different documents. For example, you might reuse feature descriptions or tasks. Note that if you do not currently use a modular approach to creating content, you might still look into reviewing the way you write in order to allow modules (topics) to be reused in different user contexts.

Making the move to modular documentation requires new ways of thinking and writing, and these take a little time to adapt to. Remember to give your team proper training and an adaptation period.

Creating Content Automatically

Word processing tools already automate some tasks, like generating a table of contents based on headings. However, if you get more structured, you can get a lot more than a table of contents. For example, you could create an overview, summary, or glossary. Figure 1 shows the description of an API class, most of whose content was created from its methods documentation.

In order to implement a solution that supports your need for automatic content reuse, you first need to define what content you want to reuse, then define and use tags in your XML information model to accurately identify that content. This process requires extra planning when you are first defining your XML content model, but, as you might have guessed, it is worth the effort.

Using Conditions and Filtering

Conditional processing enables you to define what appears or does not ap-

pear in a deliverable. The most common conditions are based on the type of user, product variants, product customization, and operating system. Many of you already use this feature in traditional tools like *FrameMaker*. But how does it work in XML?

Conditions are most often specified in attributes, as in the following example:

```
<p>This paragraph is available to everyone.</p>
<p audience="sysadmin">This paragraph is only available to system administrators.</p>
```

No matter what your chosen structure is, make sure a proper mechanism for identifying conditions and filters is in place, if needed.

What might be displayed differently?

Separating content from presentation is the best known advantage of XML. However, you can't apply any special format to information that is not identified as different. For example, if you want notes to be displayed differently from other paragraphs, you'll need to identify the note content, as in this example:

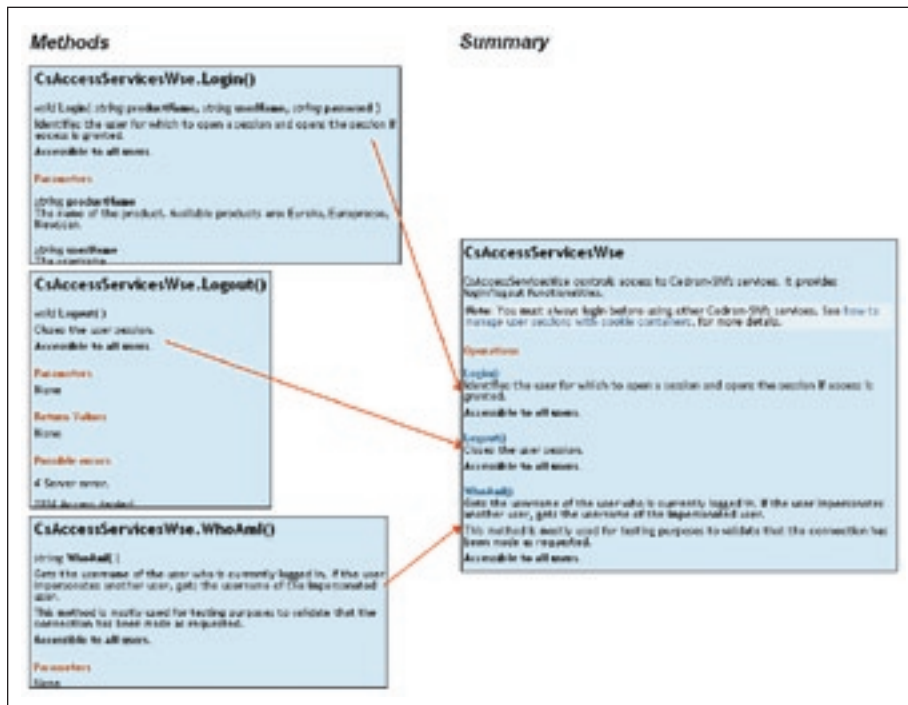
```
<p>Regular paragraph.</p>
<note>Special paragraph that may be prefixed by "Note: ", be prefixed by an icon, have a different background color, or else. The look will be defined at output generation. </note>
```

What supports workflow, document management, or even process automation?

Workflow, for some, means following the document life cycle with status (*draft*, *in review*, *approved*, *deprecated*, etc.). For others, it means the ability to assign documents to specific authors, subject matter experts, or reviewers with or without e-mail notification or tracking of the localization process.

No matter what your specific workflow includes, don't miss out on the fact that you can use elements and attributes to track and support your documentation

Figure 1. Description of an API class.



tent best, you have a lot to contribute when it comes to defining objectives for content structure, reuse, workflow, and processing, as well as automatic content generation. Make sure that everything is in place for your team to get the most benefits from an XML implementation. The advantages don't come just from technology; in fact, most of them come from having a great content model that can then be manipulated by the technology. ❶

France Baril has a unique background, with a B.A. in communication from the University of Ottawa and a B.Sc. in computer science from L'Université de Sherbrooke. She is a documentation architect at Ixiasoft, where she helps various clients enhance their documentation processes with XML-based solutions. France is president of the Montreal Chapter STC, and a founding member of the DITA technical committee at OASIS. You may reach her at france.baril@ixiasoft.com.

process. These elements and attributes are also known as metadata. They contain information about content that is usually not displayed to end-users.

Identifying what metadata to include in your DTD/XML schema is as important as planning the content structure of your XML information model.

The most widely used metadata are meant to track document status, versions, and owners; assignments to specific users; dates of creation and modification; people who modified the document; and the localization process. Note: If you use a content management system (CMS), it might help track some of these elements for you.

Where to put this information depends on your approach. If your XML files are modules or topics, you may track metadata and automate processes at that level. No matter what your decision, taking time to analyze your needs and processes and how to support them in the system always pays off.

What might be searched?

Search is a useful feature to offer end-users, and it can also be very handy for authors. Yet it is one of the most overlooked XML features.

Because information in XML is tagged semantically, search capabilities surpass the keyword search offered in traditional tools. Anything that can be tagged, like specific content units or metadata, can be found. For example, you could look for anything that you modified in the last month and that contains the word *warning* in a <note> element.

What it comes down to is that if you want to be able to find specific information, you should make sure to create elements or attributes for it in your XML content model. As with workflow, if you decide to use a CMS, it might track some of the metadata without needing to have it included in the XML document itself.

The Technical Communicator's Role in an XML Documentation Team

As a content expert, you should be able to work with developers to specify your needs in terms of content creation and manipulation. And if you thought you were in danger of being left out, I hope you now see that your help is required at many interesting levels, especially content architecture. In fact, because you probably know the con-