14 Structured authoring with XML—the next big thing

What's in this chapter

- What is structured authoring?
- What is XML?
- The impact of structured authoring and XML on writers
- Does structured authoring work with single sourcing?

With any kind of product or service, there is always something that will be "the next big thing." For example, TV viewers have heard for years about the possibility of watching movies on demand by selecting films from an on-screen menu. That service is now available from many satellite and cable television providers.

Things are not that much different in technical writing—new, more efficient processes are always being developed. You just read about single sourcing in the previous chapter. The latest emerging trend in technical communication is structured authoring with XML. This chapter explains structured authoring and XML, and it describes how they affect the documentation process—and you as a technical writer.

What is structured authoring?

Structured authoring is a publishing workflow that defines and enforces consistent organization of information in documents, whether printed or online. In traditional publishing, a style guide lists content rules, and an editor reviews content to ensure the information conforms to the approved styles.

A few simple examples of content rules are as follows:

- A heading must be followed by an introductory paragraph.
- A bulleted list must contain at least two items.
- A graphic must have a caption.

In structured authoring, these rules are captured in a structure definition document. Writers work in software that *validates* their documents; the software verifies that the documents they create conform to the rules in the structure definition document.

Consider, for example, a simple structured document—a recipe. A typical recipe requires several components: a name, a list of ingredients, and instructions. The style guide for a particular cookbook states that the list of ingredients should always precede the instructions. In an unstructured authoring environment, the cookbook editor must review the recipes to ensure that the author has complied with the style guideline. In a structured environment, the recipe structure *requires* the specified organization.

NOTE: As a new writer, you generally don't need to know how to create or modify a structured definition document, which can be quite complex. Instead, you just write content based on the rules established by the document.

Elements and hierarchy

Structured authoring is based on elements. An *element* is a unit of content; it can contain text or other elements. You can view the hierarchy of elements inside other elements as a set of nodes and branches.

Elements can be organized in hierarchical trees. In a recipe, the ingredient list can be broken down into ingredients, which in turn contain items, quantities, and preparation methods, as shown in Figure 39 on page 216.

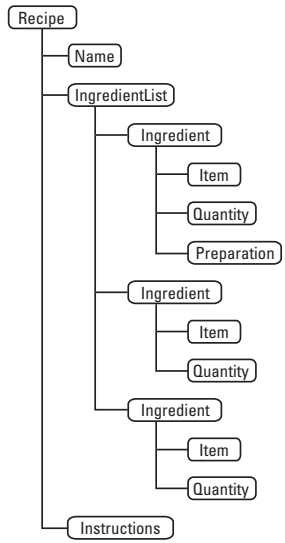

Figure 39: *Recipe hierarchy*

The element hierarchy allows you to associate related information explicitly. The structure specifies that the IngredientList element is a child of the Recipe element. The IngredientList element contains Ingredient elements, and each Ingredient element contains two or three child elements (Item, Quantity, and optionally Preparation).

In an unstructured, formatted recipe, these relationships are implied by how the type looks—for example, the recipe name is in large bold type, and the ingredient list items are in smaller type. The publishing software, however, does not capture the hierarchical relationship between the recipe name and the ingredient list—or the relationships of the components that make up each ingredient list item (item, quantity, and preparation).

In structured documents, the following terms denote hierarchy:

- Tree—The hierarchical order of elements. (By the way, it's not unusual to think of a family tree when considering the hierarchy of elements.)
- Branch—A section of the hierarchical tree.
- Leaf—An element with no descendant elements. Name, for example, is a leaf element in Figure 39.
- Parent/child—A child element is one level lower in the hierarchy than its parent. In Figure 39, Name, IngredientList, and Instructions are all children of Recipe. Conversely, Recipe is the parent of Name, IngredientList, and Instructions.
- Sibling—Elements are siblings when they are at the same level in the hierarchy and have the same parent element. Item, Quantity, and Preparation are siblings.

Element attributes

You can store additional information about the elements in attributes. An *attribute* is a name-value pair that is associated with a particular element. In the recipe example, attributes might be used in the top-level Recipe element to provide additional information about the recipe, such as the author and cuisine type (Figure 40).

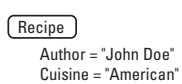

Figure 40: *Attributes capture additional information about an element*