Virtual Documents: The Challenges of Chunking

December 4, 2009

Originally published January 1, 1999 in Web Review magazine.

As we enter 1999, I have a warning for information architects around the world. Beware the virtual document! It may look harmless. It certainly looks helpful. It will lure you with a siren's song of reusable content components that enhance flexibility and improve efficiency. And then, if you're not careful, it will smash you into pieces upon the rocky shores of complexity.

Perhaps I exaggerate a little, but I do so to make an important point. Successfully integrating the concept of virtual documents into the design of Web sites and intranets is a *really hard thing to do*. Architects who fail to recognize the power of the traditional document for both users and authors are destined for trouble.

What is a Virtual Document?

Virtual documents don't really exist. Or do they? When information architecture design sessions devolve into metaphysical debates about which documents do and don't exist, you'll know you've passed through the looking glass into the world of virtual documents.

A virtual document is a collection of content chunks that are dynamically assembled from a variety of sources into a visual container that a user experiences as a document. Make sense? As you begin to see, it's rather difficult to define them in a way that makes sense to humans.

However, a simple example should help. Consider a news article intended for display on your Web site. When you present this article, you may want to include a global navigation bar at the top of the page and a copyright notice at the bottom. Since both the navigation bar and the copyright notice may be used on hundreds or thousands of pages on your site, there's an obvious value to storing each as reusable "chunks" of information that live in only one place. So, what the user experiences as a single document is really a "document instance" which consists of the article, the navigation bar, and the copyright

notice.

So far so good. The problem begins when we go one step further down the slippery slope of complexity and consider breaking the article itself into chunks. Perhaps there's a paragraph that could be reused in another article. Perhaps there's a chart that also appears in the company's annual report. Suddenly, you find yourself in the midst of a chunking frenzy, defining rules for breaking down all documents into lots of tiny, reusable content components.

At this point, you've gone too far. The promise of reusability has obscured the goal of usability. You've ignored two of the most important components in your system: Users and authors.

Users

As you're determining how finely to chunk your content, there are two primary and highly related user considerations: Size and context. A document provides context for the sections and paragraphs within that document. Break that document into small isolated content chunks and you sacrifice that context. Keeping the balancing act between size and context in mind, consider the following guidelines when determining your content chunking strategy.

- Start by acknowledging the reality of multiple levels of granularity (different sized chunks). There's a big difference between a single article and an entire newsletter. Enable users to distinguish between apples and oranges. For example, index documents according to document type (e.g., article, newsletter) and leverage that indexing in results displays.

Sample Results Display: *Your search on "virtual" returned the following 3 documents*	
Document Title	***Document Type***
Virtual Reality Today	*Newsletter*

What is Virtual Sex?	*Article*
Announcing Virtual Document Support	*Press Release*

- Size for reading both online *and* offline. Users who read online don't want to waste time or strain their eyeballs. They'd prefer small content chunks that answer one question about one subject for one purpose. On the other hand, users who print documents and read them offline may be much more comfortable with long documents. They certainly don't want to be forced to print 20 separate content chunks just so they can read an entire article. If you do break documents into multiple chunks, consider providing an easy option for printing the entire document.
- Embed context. If you break documents into chunks, it's critical to preserve the context. XML excels in this, since it provides built in context sensitivity by modeling documents and chunks within hierarchical containers that can be leveraged to improve text retrieval. For an example, try a simple search within the **HTI American Verse Project**. While the content is tagged to the level of individual lines of verse, results are presented as lines of verse within sections within poems within poetry collections.

Authors

While the needs of users should outweigh the needs of the authors, authoring quality and efficiency is obviously very important. Asking authors to write small self-contained content chunks that may be reused in multiple documents by multiple audiences is a rather tall order. While this modular approach may work for simple content chunks such as the copyright notice, authoring content is generally not like manufacturing widgets.

Over the course of a lifetime, authors have learned to write documents (not chunks) and usually with a specific audience in mind. That is not to say that you can't teach an old author new tricks, but don't expect miracles overnight. One option is to allow authors to continue to write documents as usual and to have a separate process for evaluating the extent to which those documents can be broken down into reusable chunks. You'll save the authors a great deal of grief, and perhaps develop a more realistic expectation about

the reusability of your content.