

# against *the* wall:

By Hans Fenstermacher

## content from the user's perspective

Ever wonder which wall they're talking about in the "over-the-wall" syndrome? You know, the idea that technical writers create content, then lob it over the wall to localizers, who do whatever it is they do. I know what you're thinking: It's the wall of the silos that technical writers (or localizers, depending on your perspective) live in. Yes, but there are also fundamentally more troublesome barriers here. That's because writers spend a lot of time building walls into content, not just around it.

Quite apart from the difficult workflow problems inherent in the over-the-wall syndrome, the walls writers build into technical content are a big problem. They create multilingual development issues, raise production costs, and increase time to market. But that's only our own side of the story. Let's consider (for a change) how end-users are affected by these walls, for, make no mistake, the barriers are just as good at blocking them out.

### QUANTITY 1: QUALITY 0

Yogi Berra once said, "No wonder nobody ever goes there; it's always so crowded." What he meant, in inimitable Berraesque fashion, was exactly the opposite.

The crowd is so big that it's blocking access, so people can't get in. Technical content is like that, too; words crowd every line, screen, and page so much that users can't "get in" to discern meaning, for example:

The first step in creating a drawing from an existing part or assembly is to create a new drawing XYZ file by either clicking the New icon from the What To Do section of the Getting Started page and then clicking the Default.xyz icon from the Default template tab (as shown in the following image), or clicking New from the File menu, or clicking the down arrow on the New icon from the left side of the Standard toolbar and clicking Drawing.



For content like this, users aren't going to wait in line for hours to "get in" (we're not talking trendy hip-hop club here). And if users don't get in, whatever they're looking for might as well not exist. This profusion of words, or information or what-have-you, creates an impenetrable wall between the product and

the end-users that even high production values can't overcome.

So, how do we content developers deal with these walls? Instead of trying to lower them, we rearrange the content bricks. We develop ways to reuse as much as possible (single-sourcing); we create systems for storing and managing (content management); and we apply tools for multilingual streamlining (translation

Hans is founder and president of ArchiText Inc. ArchiText specializes in localization, globalization, and end-to-end global content solutions based on ABREVE®.

For more information on ArchiText or ABREVE®, please contact:

Chad LaCroix, Account Strategist

Tel. +1 978.409.6112 or toll-free 1.800.688.7205

chad@architext-usa.com

<http://www.architext-usa.com>

memory). These approaches don't lower the barriers of access to content, but they do make the work of building content cheaper and faster. Users still aren't getting in, but the wall that keeps them out is now prettier and more efficient. (Note to content bricklayers: Put down your trowels and pick up a sledgehammer.)

#### **STRUCTURE: OSTENSIBLE CONSISTENCY, FUNDAMENTAL CONFUSION**

If users somehow penetrate the wall of quantity, they are confronted with another barrier in the form of the content's structure. Technical writers all too often allow content to be dictated by the tools and structure instead of the other way around. Consider the abstruse "stem sentence" requirement in many authoring systems. This mandates that headings be followed by an introductory sentence before the actual topic begins, resulting in absurdities like this:

### **Configuring Your Network**

This section describes how to configure your network.

To configure your network, ...

Is there a sentient person who cannot tell from the heading what this section is about? Is there a user on Earth who cares what a stem sentence is? Even if the heading were not clear (and, unfortunately, unclear headings abound), this stem sentence does nothing to improve clarity. The user just spends valuable effort learning that he is annoyed. Just another obstacle in the user's way, but the unremitting demands of the "system" are met and the writer's predilections are satisfied.

Users are intensely concerned about not wasting their time. Whole schools of product design are built around this idea, and yet, technical writers allow themselves to be forced into erecting monumental time-wasting walls by their tools and templates. Writers often become devotees of certain tools, then develop complex templates that faithfully project an unswerving consistency. The result is lovely content architectures, but not much more:

### **Installing Windows NT Server 4.0**

During this section of the procedure you are required to install Windows NT Server 4.0 on the PC that is to be used as the LAB Map Server.

#### **To Install Windows NT Server 4.0**

1. Install Microsoft Windows NT Server 4.0

*See your System Administrator for more information about installing software on your system.*

As you can see, rigorous adherence to these particular templates creates a satisfactory experience for everyone involved - except the user. This is well beyond an annoying waste of the user's time; it is a huge usability problem. Why would technical writers knowingly create a usability problem? They wouldn't, but by viewing templates as an end in itself, instead of a means to an end, this wall goes up brick by brick nonetheless.

#### **IT'S THE USER, STUPID**

In the content development community - technical writers and localizers alike - we do, however, constantly seek to improve content. We strategize on how to make the content deployment process more efficient and streamlined; we create systems for storing and delivering content; we develop tools and technologies to lower content development and localization costs. Unfortunately, we tend to approach this improvement effort from the wrong perspective and at the wrong point in the process.

Technical content exists because technical products are flawed. If the products were perfect, no

*“What if we started from the premise that the ideal amount of technical content is none?”*

one would ever need a user manual or on-line help. Since we content developers create something that is “helpful,” we imbue it (and ourselves) with a sense of importance and necessity. So, we tend to view the existence of user content as a given. Our analysis of how to improve it inevitably focuses on what we do to and with content. *Our* methodologies, *our* workflows, *our* tools, *our* metrics.

This navel-gazing has led to some improvements, but we will be hard pressed to produce major gains in usability because the concepts underlying our thinking - like quantity and structure - are self-perpetuating. Volume begets more volume; structure reinforces itself at every turn. Consequently, the tools and methodologies we have developed and employ are built around this self-perpetuation. While they represent a step in the right direction, Translation Memories and Content Management Systems, to a significant degree, miss the point. These and other tools deal with content after it is already in existence; in other words, content is what it is, now let's improve it.

Inevitably, we fail to question this view as anything but inherently good and right. But suppose it isn't. What if we started from the premise that the ideal amount of technical content is none? Viewed this way, achieving success would put a lot of us (including myself) out of business. But this is precisely the user perspective. Users don't purchase products so they can read documentation and help. They purchase products to accomplish goals - their own goals.

Now content isn't going away any time soon (whew!), but if we want to be really successful in terms of usability, we need to internalize this viewpoint and start producing content accordingly. And the best time to improve content is before it exists. After all, content doesn't just happen; each and every single word, graphic, format, font, and so on is chosen by someone. At that moment, usability begins. Where it ends will depend on how well we all question everything down the line and whether we make deliberate, well-informed choices.



#### **ABOUT THE AUTHOR**

Hans Fenstermacher is President and founder of ArchiText, a language service provider. He has developed ABREVE, a methodology for reducing volume and globalizing content.

Every other month, Hans will share his insights into the link between authoring and localization in CSN magazine. Contact Hans at [hansf@architext-usa.com](mailto:hansf@architext-usa.com).

**ArchiText developed ABREVE® in response to an expressed need for new ways to reduce localization costs while preserving the integrity of content. ABREVE® addresses these issues directly while improving content usability.**

**ABREVE® is a patent-pending methodology that enables low-risk step changes that can provide drastic localization cost reductions.**

# Architext®

## End-to-End Localization Solutions

Meeting Your Requirements for...

- ★ Lower Direct Localization Costs
- ★ Faster Time to Market
- ★ Higher Quality and Usability
- ★ Compatibility With Your Workflow
- ★ Less Investment in Tools & Training

with

# ABREVE®



[www.architext-usa.com](http://www.architext-usa.com) 800.688.7205