

# The Future of Online Help Lies Behind Us

by Rob Houser

**Note:** *This article originally appeared in the Tekom newsletter in November 2006. Tekom is a European company that provides training, publications, and an annual conference for technical communicators.*

To see the future of online help you don't really need to look forward—you need to look backward.

Consider this... Aside from the fact that nobody had done it yet, the World-wide Web didn't appear from nothing. It was based on existing technology—computers, the internet, electronic information, and standard programming languages. The innovation was in combining these existing technologies in a creative way to solve a problem—sharing information.

Theoretically, we should be able to envision the future of online help by reviewing the current technology and the current problems we face as help authors. We may have to engage in a little creative speculation, but we'll save that part for the end.

## What are the problems we face today?

- Users are too busy to read, especially help that is irrelevant to them and separate from what they are already doing.
- Users are too spread out. It's hard for them to help the person across the hall when they're working from home or in different countries.
- Users don't receive as much training as they have in the past, especially not first-hand, instructor-led training.
- Users and technical writers have been disconnected from each other for a long time and their connection has mostly gone in one direction—from the writer to the user.
- Products are highly customizable and configurable. It's hard to write about a product that has as many variations as it has users.
- Organizations have too much information to manage and the use of this information is often inefficient.
- Writers have been so consumed with keeping up with tool-based knowledge that they sometimes lose focus on their real jobs—to help users accomplish their goals using our products.

## So what is the existing technology available to us?

- XML is finally becoming the standard language for the Web, not just for documentation but also for software applications. Web-based applications are blurring the domain of developers, designers, writers, and trainers, who are all trying to deliver information through the product. User assistance is increasingly embedded in the product through text in the user interface, enhanced error messages, and just-in-time training (coaching).
- Standards such as XML, DITA, and SCORM are being supported by the tool vendors and are making it easier to connect and share information such as user documentation and training materials.
- Object-oriented programming has impacted the way we think about information. We now can consider reusable objects of information that are dynamically transformed rather than a static page that can only be presented in one way. Single-sourcing continues to evolve to give help authors more control over the information they create, manage, and deliver.
- Users are creating their own content all over the web through wikis, blogs, user forums, and tools that support user reviews and feedback such as Amazon.com. Communication with users can now work not only between writers and users but between users and other users.

How might these existing technologies solve our current problems? Here are some predictions.

### **Building Communities of Users**

Users will be able to customize their help content across their community. If Company X buys a tool, then the Company X employees will customize that help to include their own internal policies and procedures. They might also add tips and tricks to the help for the larger community of users across companies. Users will be able to rate new content and content providers to help each other identify useful information. The help authors will promote, augment, and consolidate the best new content and migrate that content to future releases of the product.

### **Consolidating UA in the Product (Interface)**

More and more "help" will appear in the product. Currently, information architects and developers are upstaging help authors by designing and often writing the user assistance that appears in the product. The problem is not that they want to write this information; it's just that they don't totally realize that the UI "is" the help. They still think of it all as the user interface, which has traditionally been their domain. Help authors need to be more proactive in determine where the help appears in the UI so the information is in the right place at the right time for when the users have a question. The tools also must converge more, so the help authors can edit the field labels and instructions that appear in the user interface without touching the developer's code. And traditional help authoring tools much provide more methods for delivering smaller chunks of information to the user interface rather than more traditional, page-based help topics.