Instilling Passion in Our Readers
How Can the Written Word Interact with Use and Usability?
ComponentOne Doc-To-Help 2009 has everything you need to author, edit, and publish documentation your end-users will love.

★ Author: Create content using your favorite editor (Microsoft Word, FrontPage, Adobe Dreamweaver, or Doc-To-Help’s new built-in XML-based editor).

★ Edit: Enjoy the convenience of visual project management features, such as a topics organizer, related topics editor, and drag-and-drop linking. Plus, save time with automatic Table of Contents, Index, and Glossary generation.

★ Publish: Generate a combination of outputs with just one click (online Help, platform-independent Web content, and press-ready manuals) without changing or reformatting content.

This unique combination of editing, single sourcing, and publishing technology translates to significant time savings and the highest level of quality. Since Doc-To-Help is the complete documentation solution, your need for additional tools has been eliminated.

Get a free trial! www.doctohelp.com
Key Authoring Features:
- New Built-in XML-based Editor
- Ability to work directly in Microsoft Word, FrontPage, and Adobe Dreamweaver
- New Microsoft Word to XHTML converter
- Content variables (reusable content “chunks”)
- Ability to add multimedia (e.g. ComponentOne DemoWorks™ Flash files)
- Style sheet / CSS editor
- Team Authoring Support
- Adobe RoboHelp, HTML Help, and WinHelp converters
- New Reusable project settings

Key Publishing Features:
- Standard Help formats (HTML Help, WinHelp, Help 2.0, JavaHelp)
- Best Web-based Help available (ComponentOne NetHelp®)
- Press-ready manuals (PDF or DOC)
- Storable output settings (themes, conditional content, custom TOC, and more)
- New Embedded Help for .NET applications (ComponentOne DynamicHelp™)
- Error detection and warning

Add DemoWorks 2009 for more multimedia!
ComponentOne DemoWorks™, a great companion to Doc-To-Help, gives you the power to produce professional quality demonstration videos without multimedia expertise. You can record any screen activity and produce video in formats such as animated GIF and Flash. Unique frame-by-frame video and narration editing makes one-take demo creation a reality.

ComponentOne  Doc-To-Help Sales 1.800.858.2739  or  1.412.681.4343
©1987-2009 ComponentOne LLC. All rights reserved. All products and brand names are trademarks and/or registered trademarks of their respective holders.
 FEATURES

 Usability

 4 Returning Language to the Spotlight: The Interdependence of Usability and Words
  In response to a shift from language to tools in the profession of technical communication, Marnell discusses the necessity of tackling the issue of usability through a renewed focus on words and “getting the message across” as clearly and easily as possible.
  By Geoffrey Marnell

 7 Making Content Understandable: Inherent Usability in Plain Language
  Haller discusses the importance of plain language in government documents and provides examples of creating usable information. He also discusses the Brayley Bill, which seeks to have all government forms and documents written in clear and concise language.
  By Thom Haller

 10 Finding Usability in Workplace Culture
  The writers were tasked with creating a content management system for a large office. After building the CMS, observing the employees using it, and discussing with them how it worked well and where it fell short, they realized the significant role that workplace culture plays in usability.
  By Jacob E. McCarthy and William Hart-Davidson

 13 Usability in Context-sensitive Help: Re-imagining the Ordinary to Provide More Business Value
  An often-neglected area of help, context-sensitive help can be a useful tool in describing specific topics in great detail. Putkey argues that when this sort of help is utilized correctly, it is likely for customer expenses to go down and the documentation be of more value.
  By Thersa Putkey

 16 Bringing Help to the Forefront: Strategies to Increase the Usability of Your Software User Assistance and Your Product
  Online help has the potential to make products, and thus users, more successful. An elegant and effective approach to online help is embedded help, and Bleiel takes the reader through the steps to create this useful assistance.
  By Nicoletta A. Bleiel
Writing Questions That Are Easy to Answer

Have you ever been asked a question on an online form that didn’t quite make sense? Or a question that seemed irrelevant? Regardless of the design and formatting of an online form or questionnaire, the wording and precision of the questions play a key role in obtaining the information you want to garner.

By Caroline Jarrett
Returning Language to the Spotlight:

Like relativity in physics, usability in documentation is a concept that simply can’t be ignored. It colors—or should color—every decision we make in designing and writing documentation. But like relativity, pinning down a useful definition of usability is no easy matter.

The International Standards Organization describes usability as “the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction.” This gives some conceptual traction, but it lacks the necessary concreteness to make it immediately applicable.

A more concrete definition, and one more widely discussed, is based on the work of Gretchen Hargis and her colleagues in defining quality documentation. This view of usability (and quality documentation) suggests that the information in documentation must be:

• easy to find,
• easy to understand, and
• easy to apply.

The linking concept here is provided by the common definition of quality as
“fitness for use.” Obviously, if a document is not fit for use, it lacks usability; if it is fit for use, it has usability—at least to some degree. And it is plausible to judge that degree on how easy the information in it is to find, understand, and apply.

For information to be easy to find, there must be sufficient signposts in places where readers are likely to look. The two most likely places are an index and a contents list, with an index arguably the more important of the two in a document of more than a score or so of pages. A contents list, though useful to the occasional browser, simply doesn’t have the degree of granularity needed to help the typical reader: the time-poor, deadline-harassed person needing to know in a hurry how to do the particular task at hand. For such a reader, a lengthy document without an index would be deficient in usability.

Other aids in helping users find information easily include running headers and footers, cross-references and hyperlinks, lists of related tasks, bread-crumbs, and a full-text search facility (especially one that enables wildcard searching and Boolean filtering). All these features, to varying degrees, help readers find the information they are after, and thus contribute to the overall usability of documentation.

Once a user has found the information they are after, they need, of course, to be able to understand it. This is where language and usability intersect. The influence of the former on the latter is the main topic of this paper. I will come back to it shortly.

The other pillar of usability is that the information, once found and understood, must be easy to apply. To that end, it must deliver what it promises. A procedure promising to explain how to set a timer on a video recorder might be easily found and well-written, but if it doesn’t fully explain how to set a timer, and under all likely conditions, then it is less than maximally usable. Moreover, it should not cause the reader to back out of the procedure by introducing prerequisites in the steps rather than in the preamble, nor cause the reader to consult other sections of the user guide in order to complete the procedure they are working their way through. Thus the information must be relevant, accurate, comprehensive, and self-contained.

But let’s return to the second pillar of usability: the information presented in documentation must be easy to understand. This is arguably the most important facet of usability in the documentation field. There may be a plethora of signposts directing a reader to the procedures they might need (and thus the information is easy to find) and each procedure may well cover all conditions and be self-contained (and thus score not too badly on the easy-to-apply scale), but if a reader has to struggle to understand the information presented to them, then the usability of the document is undeniably deficient.

But what is meant by easy to understand?

Understandability and Readability

One often hears the KISS principle extolled in technical writing circles: Keep It Simple, Stupid. Alas, the KISS principle is, to quote Shakespeare, “hoist with its own petard.” It is just too simple to be of any use. Still, much effort has gone into providing simple measures of understandability, measures that, unlike the KISS principle, have some prima facie claim to scientific rigor. These are the so-called text-based readability formulas, the most well-known of which is the Flesch reading-ease formula (the math behind the readability scores generated by Microsoft Word).

For a start, readability and understandability are often used interchangeably:

Readability means understandability. The more readable a document is, the more easily it can be understood (Samson).

Hence, readability formulas such as the Flesch reading-ease formula can be considered contenders for determining the usability of documentation (or at least that component related to ease-of-understanding).

But the Flesch reading-ease formula errs on the side of KISS-like simplicity. It takes just two features of text as its inputs: average sentence length and average syllable count. Nothing about the reader is included, such as their familiarity with the concepts discussed. And many features of text that necessarily contribute to, or detract from, understandability are ignored: conventional grammar and punctuation, typographical cueing, contradiction, inconsistency, non sequiturs, ambiguity (especially that resulting from the use of transitional vocabulary), and many more. It is just far too easy to concoct a difficult, or even nonsensical, piece of text that scores well on the Flesch reading-ease formula. Short sentences and monosyllabic words do not equal understanding.

To those who accept these limitations but argue that the Flesch reading-ease formula is still the best proxy measure of readability (DuBay), we can retort that best does not imply good. At one time, the best way we had of estimating the number of stars in the universe was to look at the night sky and count them. But that, obviously, was not a very good technique. Further, numerous studies have failed to reproduce the sort of validation correlation that excited Flesch—the correlation between Flesch scores and scores on independent comprehension tests—and any such correlation is necessarily inflated by ineradicable sampling bias.

We should not be fooled, then, into thinking that its use in Microsoft Word gives the Flesch reading-ease formula the imprimatur of scientific rigor. The formula is overly simplistic and offers little guidance in determining whether a piece of text meets any likely usability criterion.

Understandability and Communicative Efficiency

We get closer to an understanding of understanding if we reflect on why we write—namely, to communicate. We communicate if we get our message across. But our success in getting our message across can be judged in degrees. We might achieve effortless communication: our readers get our message immediately, without any cognitive or emotional struggle. At the other end of the spectrum, we might fail completely: ambiguity, vagueness, conceptual denseness, and a host of other factors might block all attempts at deciphering our intended message. And in between are the readers who eventually work out
what we mean, but only after some degree of struggle, or an encounter with more words than were necessary to get the message across.

Communicative efficiency captures the notion of ease-of-understanding far better than sentence length and syllable count. Efficiency entails effectiveness: obviously we need to get our message across if our communication is to be efficient. But it also entails that we get our message across with the least effort on the part of our readers. In other words, we should write with maximum economy, using language that is most familiar to our intended audience and that has the least potential for distraction (which might arise, for example, if we engage the emotions of our readers with paternalistic or insensitive language, or if we use language inconsistently).

**Usability, Words, and the Shifting Spotlight**

Ease of understanding, and thus usability, depends, then, on our writing exhibiting clarity, economy, familiarity, neutrality, and consistency. Therefore, it is impossible in our field to achieve maximum usability without a preeminent respect for language and for the words that are its building blocks. We risk failing to get our message across if a careless choice of words leads to ambiguity, vagueness, bafflement, offense, or cognitive overload.

Words, then, should be at the center of our professional concerns. And yet words and language can often seem of marginal concern to technical communicators. The threads on discussion forums, the articles published in our journals, and the marketing materials designed to attract students to our university courses lean strongly toward tools, methodologies, and practices. Issues of language are often missing or downplayed.

The fact that many names have been applied to our profession that do not immediately connote the pre-eminence of language may have contributed to the drift away from appreciating the importance of words. We were all once technical writers, and when we were, the importance of writing—of words and of language—was explicit. But that is not the case when we call ourselves content providers, end-user assistance professionals, information designers, and the like. Learning that someone is an end-user assistance professional would, to an outsider, give no hint that the person in question belongs to a profession where writing consumes the time of most practitioners most of time.

Moreover, some of the names we have concocted for our profession are of such bland generality as to encompass many clearly distinct professions. (A journalist, graphic designer, and musician can all be seen as content providers; and a call-center representative is also an end-user assistance professional.) We fail to identify and differentiate ourselves by adopting names that drown out our particular, unique contribution. And in doing so, we push even further from sight the fact that writing is what most of us do most of the time. We may be especially fond of tools and methodologies—and there is no harm in that; indeed some degree of tools expertise is essential—but expertise in XSL transforms, DITA, persona mapping, VBA macros, Framescript, wiki design, and the like is of no use if our writing—our particular, unique contribution—fails to achieve its primary purpose: effortless communication. It is writing before all else, and that is so even if some in our profession spend all their working time doing things other than writing.

To its credit, our profession has always prized usability. We may not have always agreed on what it means, nor given due respect to the need to clarify its definition, but a moment’s reflection on why we do what we do, on the ISO definition of usability, and on the work of Hargis and her colleagues, should bring home the fundamental importance of language to our profession. Words are what make or break us. Our technical skills are secondary and have always been secondary. Their relevance changes from year to year, version to version—unlike that of language. So if we are to continue our commendable respect for usability, we must return language to the spotlight. We must develop a passion for language that matches that of lexicographers. We must put down our prescriptive grammars and become scientists of linguistic flux. We must accept that being a users’ advocate—which most of us are—requires immersion in the users’ language. For what good is an attractive, well-structured document—even a well-crafted sentence, written once and re-used often—if it fails to deliver its meaning to the audience for which it was intended?

**Suggested Readings**


**Geoffrey Marnell** (gmarnell@unimelb.edu.au) teaches technical writing and editing in the English Department at the University of Melbourne. He is also the founder of Abelard Consulting Pty. Ltd., a documentation consultancy providing technical writing services, technical writer placement services, and training in technical and scientific writing.
Making Content Understandable:

Here’s a statement that probably doesn’t surprise you: “Plain language is usable language.” Here’s another: “Usable language is plain language.”

Surprised? Then you may be wondering about the definitions of usable. Or plain. Are you trying to figure out the difference? Does that make you feel grumpy? I’m grumpy. US citizens are grumpy. And the connections between usable language and plain language offer the key to our degrupification.

Let’s Start with Grumpy

I’m not naturally a grumpy person, but I am among those persons who are most likely to get lost in information. At first I ignored this proclivity. “I must have read it wrong,” I’d say, as I tried to figure out how I turned in the wrong direction or made the same errors over and over.

Like everyone, I’d encounter sentences that were half a page long and documents that left me wondering what the author wanted me to learn, or do. I encountered forms that gave me no help seeing which way was up. “Can’t things be understandable?” I’d wonder.

At one point in my career, I stumbled over Intranet content that was poorly conceived and crafted. I tried to articulate my challenge, but thoughts on better “usability” fell on deaf ears. “If you can’t figure it out,” I was told, “then you don’t have what it takes to work for the US House of Representatives.”

“If you cannot take the time to be thoughtful about your content and construct it in ways that help people,” I responded, with perhaps a bit too much verve, “then perhaps YOU don’t deserve to be working in the US House of Representatives.” (Some of my contracting opportunities were short term.)

Why Are We Grumpy?

In the workplace, we typically look at content so we can get something done. We focus our cognitive resources to accomplish what we want to accomplish.

But if we become burdened by the design and structure of the materials we read (rather than the intrinsic nature of the task), we instead experience extraneous cognitive load, according to the influential theory of cognitive load formulated by John Sweller, an Australian educational psychologist.

I refer to this phenomenon as cognitive overload—trying to gather meaning from a sentence but, instead of getting your job done, being bombarded by difficult sentence construction and poorly organized content. As a 20-year veteran of federal communication, I’ve experienced a culture of “good enough for government work,” I’ve clashed with the culture of abbreviations, and I’ve just about drowned in the extraneous cognitive load of government writing.

But then I wonder, if research tells us we can reduce cognitive overload and improve the ability for people to get their jobs done (which it does), doesn’t it seem important to work toward that end? Can US government communicators progress from frequently convoluted content to clear, understandable, meaningful information? I believe—as others do—they can.

Creating Change

Bruce Braley represents Iowa’s 1st Congressional District (Northeast Iowa). He recently addressed the challenge people face using government documents: “Anyone who’s done their own taxes knows the headache of trying to understand pages and pages of confusing forms and instructions. There is no reason why the federal government can’t write these forms and other public documents in a way we can all understand.”

Rep. Braley, a Midwestern Democrat with a populist streak, first introduced the Plain Language Act in 2008. It received bipartisan support in the US
House of Representatives, but the Senate never acted on the bill. He reintroduced the bill in February 2009. The bill would require the federal government to write all new publications, forms, and publicly distributed documents in a “clear, concise, well-organized” manner that follows the best practices of plain language writing.

**So, What Is Plain Language Writing?**

Plain language (originally called Plain English) is communication your audience can understand the first time they read or hear it. It’s audience-focused; language that is plain to one set of readers may not be plain to others, so authors must consider their audiences.

When we write in plain language, we structure content so people can use documents quickly and easily. Plain language helps people find what they need, understand what they find, and act on that understanding. Like good technical communication, well-crafted plain language takes the reader, the purpose, and the context into account: who is the writing for; what is the reader’s purpose for using the document; and what is the reader’s context (perhaps a sense of eagerness … or dread)?

**A Recent Example**

Consider this “disconnect” between government writing and the reader’s context. I recently purchased property in my hometown in West Virginia. A few months later, I received a letter from the county assessor:

Dear Property Owner:

In 1990, the West Virginia Legislature ordered a new statewide revaluation of all property in West Virginia. Every assessor is required to submit plans for revaluation of property in three-year cycles to maintain an assessment level of 60% of market value annually. We are now in our fifth year cycle and must revisit every property to determine if changes have occurred which would affect the valuation for the property. In order to accumulate sufficient data to achieve this goal, we have enclosed a “sales questionnaire” relative to your property. We need your help in determining if the sale represents fair market value.

Did you actually read this text? Your brain probably yelled, “SKIP THIS!” as a strategy for avoiding extraneous cognitive load. Mine did—the first time. But I was a new property owner, so I felt I should try to figure out what the county assessor wanted.

On second read, I came closer to identifying my action: “he wants me to complete a form.” I flipped the page and found a questionnaire with this warning:

Failure to return this questionnaire may result in undue increase in home value and increase tax ticket.

Tax ticket? Does that mean taxes? I assessed my next steps. As a reader of this document, I focused my cognitive resources on what I was supposed to do. And I grumbled some about the structure of the information. Why didn’t the writer consider something closer to the following?

Dear New Property Owner,

State law requires our office to collect information…. We’ve attached a form so we can update our county records. Please complete it and return it in the envelope we’ve provided. If you fail to return this questionnaire, we may incorrectly assess your property value and you might end up paying higher taxes.

I realize my on-the-fly rewrite isn’t complete. I don’t know much about the state statute, and I remain as perplexed about the levels and percentages of market value as when I first read the letter. But I believe it’s possible to adjust the text so that it is more closely aligned with the principles of plain language. And I believe it’s possible for all government offices to rethink their communication to help their readers get their jobs done.

**Clear and Usable Language**

I like this definition: *Plain language is clear and usable language. We might wonder what is meant by usable. Usability, according to the International Standards Organization, points to the quality in which people interact with a product.* Can we get things done quickly and productively? Can we get the information we need, complete our work accurately, and then feel confident that we have met our goals? These qualities of efficiency, effectiveness, and satisfaction are at the core of plain language rules and legislation.

Usability consultant Whitney Quesenbery explains that “plain language is how we make content usable.” In a conversation about “usable writing,” Whitney explained that both plain language and usability start from an understanding of human experience. “Plain language provides some basic guidelines to ensure that experience is excellent.”

You can see what “usable writing” looks like in real life. In a recent research project, Congressional staff read plain language letters in one-third the time it took them to read the original, bureaucratic versions. Look at the difference between these two:

**Original:**

If you disagree with this disallowance and believe the evidence now of record is sufficient for us to award you benefits, please refer to the enclosed VA Form 1-4107, Notice of Procedural and Appellate Rights, which explains your rights to appeal.

**Plain Language:**

If you think we shouldn’t have turned down your claim, you should write and tell us. We’ve attached a form, which explains your rights.
Performance and Transparency

Government content, especially government web content, is about performance—directing people to necessary content so that they can accomplish their goals. For example, I do not spend afternoons in my comfy chair saying, “I love to read about government grants, so I’ll go online and enjoy lots of words.” Instead I say, “I want a grant. What do I need to do? How can I accomplish what I want to accomplish? And how can government content support me?”

Similarly, people seldom want to know who works in the agencies. Instead, they want to know what the agencies do. They want content that tells a story, inspires action, or supports them in meeting their performance goals.

Obviously, government cares about performance—leaders want to see accomplishment. E-government initiatives for the past few years have been guided by performance-focused principles: citizen-centered, results-oriented, and market-based.

To support government communication more fully, the Obama administration has identified the need for a more open federal government. They have already identified three communication strategies to keep citizens actively engaged:

• **Government should be collaborative.** “Collaboration actively engages Americans in the work of their Government. Executive departments and agencies should use innovative tools, methods, and systems to cooperate among themselves, across all levels of Government, and with nonprofit organizations, businesses, and individuals in the private sector.”

  Plain language and usability research offer audience-focused collaborative methodology.

• **Government should be participatory.** “Public engagement enhances the Government’s effectiveness and improves the quality of its decisions.” Public engagement is stymied without understandable communication.

• **Government should be transparent.** “Transparency promotes accountability and provides information for citizens about what their Government is doing. Information maintained by the Federal Government is a national asset. My Administration will take appropriate action, consistent with law and policy, to disclose information rapidly in forms that the public can readily find and use.”

  Obviously, easier-to-read text helps people understand what the government is doing, and it helps prevent federal waste and abuse. Gwynne Kostin, writer, web manager, and advocate for transparency in government, expressed this sentiment in a recent online post:

> For transparency to succeed—or, in other words, for government to make sense to citizens—we need to start with the basics. I can’t participate in the debate unless I understand the choices…. Adding the call to using plain language—not legalese, governmentese, technicalese or gobbledygook—needs to be integrated into each step of the transparency process.

These administration ideals align with the research on developing understandable and usable content. Government leaders who rethink their content and incorporate plain language principles will find they have built a framework for a more usable government.

**Resources**

ISO 9241, Part 11.


**Learn More About Federal Plain Language Legislation**

The Plain Language Act of 2009 (introduced in the House) is identified as HR 946 IH. Its purpose is to enhance citizen access to government information and services by establishing that government documents issued to the public must be written clearly.

**Tracking the Legislation**

The legislation has been referred to the House Committee on Oversight and Government Reform. You can track this bill using Thomas, the Library of Congress web tool (www.loc.thomas.gov).

**Following Plain Language**

The Center for Plain Language is a nonprofit advocate for clear and usable government writing. Find information on its programs and outreach at www.centerforplainlanguage.gov.

Plain Language Association International (PLAIN) is an international community for plain language advocates (www.plainlanguagenetwork.org/).

A Facebook community, “I will no longer accept hard-to-read government documents,” is a meeting place for Facebook members who want to see change in government communication (www.facebook.com/group.php?gid=48546973325).

**Learning More About Plain Language**

The website www.plainlanguage.gov is the repository for government content on plain language principles.

Thom Haller (thom@thomhaller.com) has more than 15 years’ experience teaching adults and university students in strategies for writing clearly and effectively. Since 1998, he’s been teaching classes on developing web content and “architecting” usable web/Intranet sites. He has worked with a variety of federal agencies, associations, and NGOs, teaching and structuring information so people can get work done in a manner consistent with their job contexts. An advocate for plain language in government, Thom guided students and volunteers through the process of reshaping www.plainlanguage.gov. He served a year term as executive director for the Center for Plain Language (a nonprofit organization focused on improving the structure and writing of business and government documents). Thom also manages the Facebook community, “I will no longer accept hard-to-read government documents.”
Finding Usability in Workplace Culture

By Jacob E. McCarthy, Member, and William Hart-Davidson

Technical communicators’ work often involves questions of usability. Because technical communicators must not only write well themselves, but also help others around them write well, too, usability becomes all the more important in the design of software systems for content management. For such systems, traditional usability testing—asking users to complete a task and then noting the system’s ability to help them—may not quite give us the whole picture.

Anyone who has shared a writing project with co-workers knows that writing isn’t just about the tools we use. Writing well requires a blend of individual ingenuity and group coordination. As such, all writing tasks are shaped by the culture of the office environment where we do our work. Simple, discrete tasks that place one person in front of a screen in the hopes of evaluating the usability and usefulness of the writing environment are usually not sufficient to determine if that tool will succeed or fail in the workplace. In this article, we share our experiences as writers creating writing software. While creating a content management system (CMS) for writers in a large office setting, we found that the innovative and culturally driven writing practices we observed were key to our understanding of when the CMS we were developing was working and when it was not. We share some of the lessons we learned about how to pay attention to workplace culture as part of usability evaluation during the design phase and after rollout.

The Project

When our research team started building a CMS for writers in an administrative office at a large organization, we set out to design something that would mirror the writing habits they had already formed. We started by interviewing the writers, asking questions about why they write, what tools they use, and how they collaborate with one another. This kind of information is useful for creating user profiles and workflow diagrams to guide the development of the software. When we learned that writers often used email to share documents, we built tools into the CMS that would provide a similar function. By building a system that supports rather than changes or replaces the current work habits in an office, the transition to a CMS can be made easier for the writers. Even with this kind of careful planning and attention paid to how people work, some type of usability testing is necessary to know whether technology is living up to its potential.

When considering how best to evaluate our CMS’s usability, traditional usability testing didn’t seem like the right
way to go. For starters, we didn’t have the facility to perform a laboratory test or the budget to hire that work out. Also a factor, though, was the fact that we already had a wealth of data about the writers’ work that we collected prior to development using qualitative methods such as interview and observation. Our best data had come from meeting with writers in their workplaces, and we wanted our usability data to mesh neatly with those earlier findings. By comparing the two, we hoped to be able to figure out not only how to make the CMS better, but also how it changed the writers’ work habits. A usability testing method based on interview and observation made the most sense for our project.

Over several months, we continued to meet and watch writers in their work environments, taking note of how they used the CMS and traditional office tools. We also interviewed writers individually, and several times met with the entire group of writers to discuss their expectations and experiences with the CMS. While not usability testing in the traditional sense, this method did offer some valuable insights into how we could improve the usability of the CMS, and it also gave us a clear image of the role the office culture, and not just the interface, played in usability. Here we’ll provide a few stories about what we observed in order to illustrate the benefit this method of usability testing had for our project.

Confusion Over Reviewing

Robert and Brett work in the administrative office for which we built the CMS. They often collaborate with one another to write employee profiles for the organization website and printed communications materials. When we originally interviewed Robert, he said the materials he writes are reviewed and approved by Brett before they can be made public. In response to this, we built a review function into the CMS. Authors may select a reviewer for their document, and the reviewer is prompted by the CMS to type a short response in a text box to the author. We thought such a system could streamline the approval process for Robert and Brett, but in a session observing Robert working with the CMS, it became clear he and his colleagues had different uses for the review system.

When we sat down to observe Robert working with the CMS for the first time, he was responding to a request from Brett that he review a document she had written. We watched as, rather than typing a brief message of approval or suggestions into the text box prompted by the CMS, Robert moved outside the CMS and created a new version of the document in Word, making great use of the track changes and commenting features. He then uploaded the annotated document to the CMS, sent Brett an email to let her know how he had delivered his suggestions, and said he would likely also follow up with her in person.

After observing Robert perform this task several times, it was clear that the text box was not an appropriate tool for Robert to share such thorough suggestions with other writers, so we asked him about this practice in an interview. We quickly learned that within the culture of this office, “reviewing” a document meant to actively collaborate in its revision, offering suggestions and checking with colleagues through multiple media to ensure suggestions were received positively. When designing the system, we used the word “review” to describe quickly verifying the acceptability of a document and offering approval, resulting in a serious ambiguity.

This lesson held several implications for our CMS design. It meant that while the function we had named “review” was necessary, “review” was not actually the best word to use, since within
the culture of the workplace, “review” meant a much more rich and active collaborative event. It also demonstrated the appropriateness of this open-ended questioning as a measure of usability. It allowed us to witness the particular work that led Robert to innovate a collaborative writing method and, because we were not asking him to perform specific tasks, we could acknowledge the usefulness of his successful innovation. Our open-ended questioning revealed that Robert usually talks in person with collaborating writers, and it made crystal clear the fact that the office culture of these writers will often dictate how they used the CMS.

**Challenging Perception of Responsibility**

Frances is an administrative support writer who frequently coordinates collaborations between multiple superiors in the office, which she says results in her work sometimes being “politically charged.” We were in the office one morning when she faced the task of synthesizing multiple editorial comments to a single document written by a co-worker. The editorial comments were from two additional co-workers and, like the review that Robert composed, they were delivered as highly annotated Word documents. Frances had an email from one co-worker on how to navigate any political dangers his suggestions to the document might pose, and she toggled between multiple browser windows to create a unified text. When she saved the new synthesized document and returned to the CMS homepage, though, she mentioned something about the CMS that bothered her.

The CMS homepage lists writers’ recent actions, a feature familiar to anyone who has used Facebook or similar networking software. For instance, when Robert reviews a document, the CMS homepage publicizes for all users that “Robert REVIEWED the document NEEDS-ANALYSIS REPORT.” Similarly, when Frances executed her co-workers’ suggested changes to another writer’s document, the CMS recorded that “Frances EDITED the document PROJECT PRE-PROPOSAL.” Frances was troubled by the way the CMS gave her full ownership of decisions made by her co-workers and merely executed by her as part of her normal job duties. Considering the potential for conflict surrounding some of the documents that cross her desk, she sensed a possibility this feature could spell trouble for her in the future.

Because it didn’t stop Frances from successfully completing her work, this problem may not be a usability issue in the traditional sense. It’s unlikely it would have been turned up by traditional task-based usability testing. It could keep her from completing other projects, though, if she needs to devote time and energy to smoothing over office-politics scuffles. And it’s also a great example of the way that potential problems with software can sometimes be attributed to workplace culture, and of the appropriateness of alternative usability testing methods to uncover these potential problems.

**More About the Method**

Our intention with the stories above was to demonstrate the utility of alternative research methods for showing how workplace culture affects usability. Keeping an eye on office politics and entrenched practices while searching for usability issues gave us a unique view of the decisions writers made while interacting with the CMS. Instead of recording the length of time and number of steps required for people to complete tasks, we focused on the cultural motives for completing tasks differently than we had anticipated. Certain traditional usability problems turned up through our observations, but our presence in the work environment made the workplace culture impossible to ignore.

With that said, there are limitations to this method of testing that mean it should not replace usability testing. Without video and audio recording, our method doesn’t provide the finely coded data that makes traditional usability testing so compelling. Similarly, our choice to observe writers at work rather than direct them through tasks in a laboratory meant we couldn’t directly compare one user to another. It did mean, though, that we could see uses of the system we had not anticipated—uses that revealed compelling and useful stories of workplace culture and usability.

**Conclusion**

By this point, the importance of paying attention to workplace culture when designing and evaluating writing software systems should be evident. When planning our large-scale CMS project, we did research that acknowledged workplace culture both before the design phase and after the rollout. We observed individuals as they went freely about their work and met with teams to experience the social dynamics in the office first-hand. An added bonus to these untraditional methods came in the way we saw users reacting to problems in the software. When encountering a design flaw or usability challenge, writers did not stop working, as traditional testing sometimes suggests they might. Rather, they used this system and others to create viable work-arounds.

By electing to gather usability data not only through traditional formative methods but also by maintaining a presence in the work environment, interviewing writers, and observing their practices, technical communicators can improve their knowledge of how workplace culture affects writing practice. That’s knowledge that may not always be available through traditional usability methods but will prove just as beneficial in iterative design stages as the quantitative data traditional usability tests provide.

Jacob McCarthy (mccar244@msu.edu) is a recent graduate of the Digital Rhetoric and Professional Writing MA Program at Michigan State University. He is also a communications manager in the Housing and Food Services Division at Michigan State University, where he focuses on internal communications with a large and diverse workforce.

Bill Hart-Davidson (hartdav2@msu.edu) is associate professor of Rhetoric & Writing and co-director of the Writing in Digital Environments Research Center at Michigan State University. He is also vice president of the Association of Teachers of Technical Writing.
You’ve probably heard or read Bob Boiko’s (founder and president of Metatorial Services Inc.) phrase, “If we deliver the right information to the right people in the right way, it will help us meet our goals.” After listening to one of Boiko’s presentations, I was inspired to think of practical ways to create useful technical documentation to meet company goals. Context-sensitive help (CSH) has long been a realm of documentation that has been neglected but, when implemented effectively, can cut down on customer support expenses and give more value to the documentation. When applying Boiko’s phrase to technical documentation and context-sensitive help, we can ask questions such as:

• Are we providing the “right information”? What is the right information for CSH? What problems are users having with the software, and what information might be needed on this context-sensitive help page?
• Who are the “right people”? What kind of user are we providing the CSH for? What type of user presses the Help button or the F1 key? What constraints are imposed on the user and what freedoms does the user have?
• What is the “right way”? How does the information need to be designed and delivered to be helpful to the user?
• What are our “goals”? What are the goals of the user, each writer, the department, research and development, marketing, support, the company?

How is context-sensitive help helpful?

Context-sensitive help is the help that appears once the user has pressed the F1 key or the Help button. When set up properly, a specific topic appears about the screen, window, wizard, or page the user is looking at to help with the specific action on the page. According to Wikipedia, “Context-sensitive help, as opposed to general online help or online manuals, doesn’t need to be accessible for reading as a whole. Each topic is supposed to describe extensively one state, situation, or feature of the software.”

Looking at a few articles on the STC Usability SIG website, we might conclude that this general theory and our general application of Help isn’t enough anymore. As software becomes more complex, users’ questions become more sophisticated and questions start to vary from the typical “how-to” procedure. In fact, after users figure out how to do something, they may begin looking for answers to the questions: Why do I need to do this? When do I need to do this? Which one of us should be doing this?

The traditional way of providing content in a context-sensitive help link has been to link to the procedure for the page. Not only can this be limiting for the user, but it also can be limiting for the writer and the company. By picking a specific procedure, the writer is guessing at what the user wants to do on the page. While usability testing would give us more insight, if we don’t have this option, we can make the context-sensitive help link more applicable to more people by providing a link to both a How topic and the Why, When, and Who information.

If we don’t provide additional options in the context-sensitive help, we are, in effect, making it more difficult for users to have their questions answered. By forcing them to browse and search through a help system they may be unfamiliar with, and by not highlighting the various options available for that screen, window, or page, we limit the options for the user and perhaps contribute to higher support costs.

How is context-sensitive help helpful?

Context-sensitive help link can improve the usability of the help.

Providing more complex information in the CONTEXT-SENSITIVE help link can improve the usability of the help.

Creating more with context-sensitive help.

Providing more complex information in the context-sensitive help link can improve the usability of the help, if not the software itself. Providing the right context sensitive help to the right user in the right way can help the company either reduce support calls or improve customer service.

As a new approach to context-sensitive help, we can provide more information in the help page in several ways:

We can provide links to related topics that allow people to scan the link to determine if it is applicable. This can help with browsing through the help.

Every page has primary and secondary tasks: tasks the user is likely to do on a page, tasks they’re less likely to do, and background information on the tasks for the page. The context-sensitive help topic can contain links to other pages as well as a reused paragraph from another topic, and a link to the basics. The primary tasks appear in the most important area in the topic (which, in North America, would probably be the top left-hand side of the help topic). Secondary information appears in a less prominent area of the screen with the basics out of the way but still accessible.

For pages that are popular or receive a lot of questions, we can provide links to the index, search, knowledge base items, workarounds, and FAQs. To find the most popular topics or search terms, we try to get analytics numbers or ask the support team for the most common questions. We can improve the more popular topics to answer questions that many people ask.

Additionally, we give them a link to search, or display the search tab along with the help topic, to open up more possibilities. If the user doesn’t see what’s needed in the displayed topic, she can immediately turn to the search box. While a writer’s life might revolve around the table of contents, a user’s life doesn’t. The structure of a table of contents may be unfamiliar to most users and, without testing the structure, we have no way to tell if it is helpful to the users. With search engines exploding in popularity, writers can take this as a cue that users are interested in searching instead of browsing.

Meeting business goals.

This new approach can help companies meet their business goals, which include reducing customer service costs or showing more value in the product and more concern for customer needs. By creating more relevant context-sensitive help and anticipating alternative questions the user has, the company
can answer more questions before the user calls in.

Updating context-sensitive help may take time and you’ll need to justify this to your company. When making a business case for improving the context-sensitive help, start by finding out the company’s goals, the department goals, and your goals. Relate the context-sensitive help updates directly to achieving these goals. Include information to show the value of helping customers by giving them better access into the help sources. Show that you are helping your company take better advantage of the existing information by linking to other items, such as the knowledge base, conceptual topics, or FAQs.

Make sure you estimate the amount of time the update will take as well as the manpower needed for the effort. If you think your company will be leery of a new project or a new effort, propose a pilot project. Ask what the top five problems are and try to address these in the context-sensitive help. The call volume may go down, the calls may take less time, the questions may be more targeted, or the user may be asking for more advanced help.

Although you can change the context-sensitive help, this isn’t a win just for technical writing but also for customer service. You can encourage your company to advertise these changes to users. We’ve all heard the refrain “No one ever reads the help.” In a market where information is a company’s most valuable asset, and where many companies spend a significant amount of money on documentation, this attitude needs to change.

But these attitudes will never change if people don’t know that information exists! Ask your company to market the information internally or externally. Marketing is always looking for a new press release or blog post, so propose these ideas and offer to write them. If a company puts money into documentation for the goal of user support, why not tell the users about it? Companies don’t traditionally think about this kind of promotion, but if there’s any way a company can stand out above a competitor, it should be promoted.

Promoting your profession (and yourself).

What does all of this get you in the end? It can give technical writing a higher profile and you a reputation of helping the company respond to customer needs and creating more value within the product and company. You have to suggest the change, but if people find the context-sensitive help more valuable, then the information you produce is more important to people. That could prove the value of your job.

As an information architect and content writer, Theresa Putkey (tputkey@keypointe.ca) always looks for more effective uses of information to meet business and consumer goals.

---

**Become a Certified User Experience Professional through our three-day workshop.**

Register now at [www.de.ttu.edu/techcom](http://www.de.ttu.edu/techcom)
Bringing Help to the Forefront:

Strategies to Increase the Usability of Your Software User Assistance and Your Product

By Nicoletta A. Bleiel, Senior Member

usability

BY NICOLETTA A. BLEIEL, Senior Member

Strategies to Increase the Usability of Your Software User Assistance and Your Product

Bringing Help to the Forefront:

BY NICOLETTA A. BLEIEL, Senior Member

Strategies to Increase the Usability of Your Software User Assistance and Your Product
Online Help became a required component in software applications in the early 1990s, driven in part by Microsoft’s “Designed for Windows” certification process. Help was generally integrated into the product from the Help menu, with context-sensitive Help buttons on dialog boxes and “What’s this?” field-level help rounding out the list of ways users could access and interact with the information.

After learning how to develop and design Help, technical communicators have been developing and discussing ways to bring it to the forefront ever since — not because of vanity, but because we realized its potential to make users (and therefore our products) more successful.

Embedded Help (in this context, I am specifically referring to an embedded pane in the user interface, or UI) is widely acknowledged as one of the best ways to integrate Help within the interface, but it has the reputation of being difficult to develop and execute. Without the ability to completely control the information and the mapping of that information to the interface itself, technical communicators faced an implementation that was expensive and therefore hard to sell internally. In addition to implementation issues, an embedded Help pane could add another online deliverable to the documentation mix—increasing the product footprint—and that time could be spent on other user assistance offerings.

But let’s put all that aside and take a look at one way to elegantly implement and map embedded help. With proper planning and design, one Help file can be developed to work as embedded assistance and as a stand-alone project.

The Case for an Embedded Help Pane

According to Galitz in *The Essential Guide to User Interface Design*, a typical user interaction with documentation involves three steps: “(users) need to find it, understand it, and apply that understanding to solve the task at hand.”

Online Help is passive, with the user taking the initiative. Embedded Help is nonintrusive and does not require the user to take the initiative, says Lynne Hall. “Embedded Help sidesteps the user’s reluctance to go to the help system, providing background and often invisible support to users.”

Embedded Help also encourages learning, because, according to Mike Hughes, “users shift from a learning/exploration mode to a task execution mode ... they stop exploring and experimenting.” Embedded Help can delay that shift.

An embedded, dynamic Help pane displays relevant information as the user navigates the interface, making the concepts and language of the interface obvious to the user. It’s task-specific, context-specific, doesn’t require users to look for it, and doesn’t require them to ask the right question. Users continue their workflow and don’t have to split their focus between two different windows, unlike traditional online help. As Cheri Zubak, one of the pioneers of embedded Help, has noted, “It’s help that is designed as software.” Best of all, users don’t see embedded Help as Help (see DeLoach).

Combine embedded Help with an information-packed Start or Welcome page and you will ease users into the interface. This solution brings Help to the forefront and, at the same time, promotes the entire user assistance set and encourages its use. This will increase the usability of your documentation set and your product.

**OK, OK, You’ve Convinced Me—Embedded Help Is Cool, But …**

I have discussed a number of reasons why embedded Help is an excellent way to increase the usability of your software application (actually, I have quoted some very knowledgeable people), but I haven’t addressed the BIG question — how is it implemented?

*Figure 1. The product interface with the embedded Help pane on the right. Note the variety of UA (traditional Help, videos, tutorials, quick reference guides, etc.) available from the Start Page.*
Traditionally, implementation has been challenging because the pane itself had to be coded, and the mapping of Help to the interface had to be coordinated between the software development and information development departments. A combination of the following file types and technologies could be used: Text and RTF files, HTML, DHTML, XML, style sheets, JavaScript, Jscript, ASP, JSP, PHP, VB, C++, and C#.

Once implemented, changes and updates could be time-consuming, and since traditional Help is still a necessary deliverable (and rightly so), two help files would still be needed.

My solution: I very nicely asked some extremely knowledgeable software developers at my company to build a control (a visual component that can be snapped into an interface and configured) to my specifications. One of my priorities was a mechanism for the information developers to control the mapping process. After an exchange of ideas, the control more than met my requirements, and it can be used in any application developed in Microsoft Visual Studio.NET. The Help file can be compiled HTML Help (a .chm) or web-based Help.

I mentioned above that traditional Help is still required. Why? Because the Table of Contents, Index, and Search are still necessary navigation aids and should be available if a user prefers to work that way. I created web-based Help for this project; web-based Help has a number of advantages over HTML Help, but one aesthetic advantage: I had greater control over the skin and could design it to match the product.

Designing User Assistance for Multiple Uses

You can use the same source files for embedded help, stand alone help, and a manual if you single-source it with a plan. It is efficient reuse and a time-saver, but it needs to be designed well.

First, choose the interface elements for which you want to provide embedded assistance. For example, if your application includes ribbons, you could choose to provide a topic for each ribbon, break it up into one topic for each ribbon group, or even one topic for each button. You could even do all three. After you have made your decision, break down the topics you’ll need to cover the entire UI. That is one set of topics. These topics should be conceptual overviews that link to more detailed task and reference information.

Another set of topics are the ones that cover the major functionality of the product. These topics will be the bones of your file and will most likely be the top-level “books” in your TOC.

Finally, you will need to list each dialog box that will require embedded help. You also have a decision point for dialogs—will you map the entire dialog box to a single topic or provide Help at the field level? Once you do that, you’ll have another set of topics. Some of the dialog topics may double as “functionality” topics.

Remember, everything that you want to map to the interface must be a topic, so choosing to map to every toolbar button and dialog box field, check box, and radio button will break your Help file into very small pieces, and logic may suffer. Any information that can be logically grouped together should be.

Once all this information is gathered, you can put together a draft TOC. Every topic that will be mapped to the interface should be flagged.

This is a good time to sketch out the relationships between topics, which will be useful later for creating “see also” links. In addition to the “see also” links at the bottom of each topic, “bread crumbs” are a great navigation aid for the top of each topic, since they indicate the user’s position in the information hierarchy and, like the “see also” links they provide clues about where they should navigate next. The navigation tools on the embedded help pane provide another level of navigation, but they will only take the user back to topics they have already visited.

Mapping and Delivering Embedded Help

Mapping the Help to the interface was handled with a mapping tool that was...
Embedded help is nonintrusive and does not require the user to take the initiative.

Figure 3. The Help pane in three phases; with the mapping interface open and unmapped, mapped, then closed and ready for dynamic action.

Figure 4. A Microsoft Word Expanded Tooltip.

Building embedded help into the control. All the mappings are saved to a single XML file that must be delivered to software development.

It was a five-step process:
1. Install the software application.
2. Drop the Help files in the folder specified by software development. I used uncompiled HTML web-based help, so I had multiple files; if I had created compiled HTML Help, I would have only needed to drop the .chm file.
3. Open the mapping interface using the key combination specified by software development.
4. Map the topics to the interface.
5. Deliver the Help files and the mapping XML file to software development for the install.

Integrating Other Deliverables, “Layering” User Assistance

Embedded, dynamic Help is just one piece of the deliverables set. Documentation needs to be layered to provide a user experience that serves all types of users and their learning styles.

In Usability Engineering (1993), Jakob Nielsen noted that because users have many uses for documentation, you might need three levels: short reference cards or job aids (i.e., quick reference guides), tutorials for learners, and traditional reference manuals for expert users. That is a good baseline, and now we can create many additional forms of user assistance that make it possible to engage visual, auditory, and kinesthetic learners, as well as different experience levels.

These deliverables include videos and podcasts, quick reference materials, and tutorials, as well as websites, company forums, blogs, wikis, Twitter feeds, and other Web 2.0 initiatives. With a little planning, these can be integrated into the interface—and not just into a Help menu (although if you have one, they should always be there, too). If your interface has a web-like design, incorporating them into the Start or Welcome page is an option. If it doesn’t, perhaps streamline a bit by creating a webpage that incorporates all of these deliverables and link to it from a menu, or from the splash screen. Links to information about training courses, consulting options, upcoming conference exhibits, online newsletter subscriptions, and other useful materials aid users, even if they aren’t documentation (and can increase revenue for the company). Links to all of these should also be included in the online Help.

One other embedded option to consider is Expanded (or “Super”) Tooltips, which can be found in Microsoft Office 2007 and other products. As you hover over toolbar buttons and other parts of the UI, these tooltips display expanded information, (including graphics) and are formatted to be easier to read. If your product includes these, you should...
be writing the text—if not, you should suggest expanded tooltips for your interface. They provide useful and timely information to users while they are exploring the interface.

**Ever-evolving User Assistance**

The constant evolution of user assistance deliverables is exciting because of the many options we have for tailoring the appropriate set for our product. And, with a little additional planning, that carefully crafted set can increase the usability of a software application. That is the ultimate goal, and it has been since online Help (and even manuals) was introduced.

**Further Reading**


Nicoletta A. (Nicky) Bleiel (nickyb@componentone.com) is an STC Director-at-Large and a senior information developer at ComponentOne in Pittsburgh, PA. She has been a technical communicator for more than 14 years; with experience designing and writing for products in a variety of industries; including information development, media sales, industrial automation, simulation, and pharmacy. Nicky has delivered presentations at STC international, regional, and local conferences and meetings, as well as DocTrain, WritersUA, tcworld, and LavaCon. Topics have included embedded help, tools and technologies, user assistance design, single sourcing, wikis, Web 2.0, and convergence technical communication.

**Considerations for Embedded UA**

- Understand single-source authoring and concepts. Take advantage of conditional text, variable text, and other options your help authoring tool (HAT) has to offer. For example, you’ll need to optimize space in embedded topics, so incorporating expandable or collapsible text (information that displays only when a user clicks on it) is a great option.
- When designing your Help, choose colors, fonts, and styles that work with the user interface. This is always a good idea, but it is especially important for embedded UA.
- Work with software development to determine the best position in your application’s UI for the embedded help pane. Galitz recommends the right side of the interface, in order to “minimize obscuring of screen content—in a window, position priorities are right, left, above, and below.”
- Once users are comfortable with the software, they may want to turn embedded help off. Alan Cooper notes that the ability to configure an interface will make a difference between liking and disliking a program. Make sure users have the option to close the embedded help window in the main UI and dialog boxes.
- No topic should be a dead end. My recommendation: At the bare minimum, every topic should have a “see also” link to all of its subtopics, and every subtopic should have a link back to its parent.
- Avoid adding requirements just for Help, such as separate databases or plug-ins (see Mueller).

Once you have worked out the design, the writing goes very quickly, but the design process can sometimes be a lot like putting together a puzzle that is mostly sky.
BY CAROLINE JARRETT, Associate Fellow

If you create a form with just boxes to fill in and no words, then it’s rather obvious that your users will have difficulty with it. It’s the written words that turn an arrangement of dots and lines into a conversation between your users and your organization.

Questions Matter on Forms

Recently, I was signing up for an online tool, “Comapping.” The process was straightforward: provide some personal details, choose the level of service, and insert a couple other bits of information, as you can see from Figure 1 below. Removing the questions, however, makes the form rather difficult to deal with.

If you go hunting on the Internet for information on creating better forms, you’ll find:
• plenty of material about the programming side of building forms;
• a few, more-enlightened discussions about styling forms using CSS; and
• the keenly debated topic of where to put the labels in conjunction with the fields.

All of these details are important. If your programming people can’t build it, then your users definitely can’t use it. If it looks ugly, your users won’t like using it. If the users can’t see which label goes with which field, they will also struggle.

But we should be paying more attention to the words: the questions and instructions that turn a meaningless selection of boxes and buttons into a meaningful conversation between your organization and your user.

A Prompt or a Fully Formed Question?

When programmers (and HTML coders) refer to “a label,” they mean whatever text will be associated with the field.

Now that I have put the words back (see Figure 2), you can see some prompts:
• First Name
• Last Name
• Email
• Password

And two fully formed questions:
• Are you 13 or older?
• How did you learn about Comapping?

Slot-in Answers Work with Prompts

Taken out of context, prompts can seem rather abrupt. The computer is almost barking its requirements at the user. But in practice, people seem to cope with them admirably well, provided that:
• The prompts are asking for information that the user is expecting to supply in this context.
• The information is readily available in the users’ head.
• The user is willing to reveal that information to this organization.

I call these “slot-in answers” because the user is more or less expecting to type them in and is just looking for the right slot to put them in.

Think about something you sign on to every day, such as your work computer or email. You’re expecting to provide that username and password, and you could probably type them in without even looking at the screen.

Once a user has decided to register with an online tool or a website, for that matter, there is an expectation that some sort of identifying information will be required. “First Name” and “Last Name” are both conveniently available in the user’s head, and a brief prompt is acceptable as a way of guiding the right information into the right box.

If Privacy Is an Issue, Add an Explanation

There is a crucial point when users must be willing to reveal their information. Many of us have learned to be guarded about our identities online, thinking carefully about which details we will entrust.

For example, “email” and “Password” in Figure 2 fulfill the first two criteria for slot-in answers: most users who are registering for an online service would probably be expecting to supply this information and are likely to have some prepared answers available. But “willingness to reveal” is a more complex matter. Would you give this information to an organization that you did not know?

This tool was recommended to me by respected colleagues on an STC project, so I typed my answers in without hesitation. However, another user might feel differently and may want to see some sort of assurance from the organization about how they protect personal data and what use they make of that data.

If you do not justify your requests for sensitive personal information, then you will find that some of your more web-savvy users will provide dubious email addresses. Less web-savvy users will simply refuse to fill out the form.

More Complex Answers Require Longer Questions

Often you'll need answers on your form that are more difficult. The upper part of this registration form has two of them:
• The unusual legal requirement to declare that you are over 13 years old
• The more straightforward marketing question of “how did you learn about Comapping?”

“This is a legal requirement” can help you to persuade your user to answer a range of questions, especially where the connection between the question and the legal ramifications is fairly obvious, such as restricting a web service to people over a certain age.

On this particular form, the designers did well to opt for a longer question and an explanation. In Figure 4, you’ll see three different versions of the same question:
• The top version is a plain prompt. It really does not give the user enough information.
• The middle one (as on the current form) does a better job, but assumes the user can guess that checking the box means “yes.”
• The bottom one uses a more explicit explanation and also avoids the slightly odd requirement of ticking a checkbox for a question where there are two possible answers.

That final version probably seems a bit wordy and formal for this particular form, but I wanted to show you that you can alter the tone of the conversation between your users and your organization quite easily by minor adjustments to the wording of your questions.

The question “How did you learn about Comapping?” provides us with an example of another type of answer—one that most users are likely to have to think about for a moment (leaving aside those who are so web-savvy that they expect organizations to ask them this type of question and have prepared answers available). I would describe this as a “created” answer: the user has to be creative to think up an answer, perhaps by remembering an immediate past experience. A fully formed question helps in these moments.

Precision in the Question Can Help Users

Asking a marketing question is not particularly harmful—if the user skips it, or provides a slightly less-than-accurate answer, no real harm is done.

But some questions really do mat-
For example, many of us these days are thinking about job application forms. I did a review of these for Socitm, a UK organization for people working in information technology in local government.

These job application forms usually ask applicants to explain why they are suitable for a particular job. Figure 5 shows three of them, in order of increasing complexity:

At first glance, there seems to be no contest about which of these questions is better: the first, simple one is clearly the least terrifying.

If we look into the problem more deeply, the second one actually gives the applicant more help, because it explains the types of experiences that the applicant is allowed to include.

The third one is hard to read, but it is actually the most precise of all of them. In local government in the UK, job applications must be assessed in a formal process that explicitly relates the details given in the application to the requirements of the “Person Specification” for the job.

The ideal question would combine the simple language of the first example with the detail of the second and the precision of the third. Here is my suggestion:

“Simple” is not always best. Think about:
- why the organization is asking the question
- how they will use the answer
- how best to guide the user to providing appropriate, usable answers

Copying Answers from Somewhere Else

Another type of more complex answer is the one where the user has to look away from the form to copy the answer from somewhere else. I call these “gathered” answers. For example:
- Credit card details, where most of us would need to get the credit card out of a wallet
- Paying a bill, where you might need to copy some details from the printed bill

Many gathered answers are almost as straightforward as the slot-in ones. To continue with the credit card example: most web-savvy users have become familiar with the three-digit code on the back of their credit card, which has such a variety of different names. I have seen all of these recently: “CVV,” “Security Code,” “Card Security Code (CSC),” “Last three digits on signature strip,” and “Signature panel code.” It has become standard for websites to offer help on how to find these codes, and the best of them (as in Figure 7) include a picture that helps you to locate the answer.

Contrast this with the excerpt from a government website, in Figure 8.

The site includes a help icon: the small green question mark in a circle. Unfortunately, the help provided is useless (see Figure 9).

Unusual Questions Need More Explanation

Users have a strong sense of what they expect to provide as answers in most transactions, as we saw in the registration form above. Here is an excerpt from another registration form; we have already dealt with name and email address.

The site has opted for a prompt rather than a fully formed question. But what do they mean? Is it birth country or country of current residence? For many of us, those will be the same; but for many others, it is not at all clear. And why do they need it? This question needs a longer explanation, both of what they mean and why they want to know.

The best way to find out what users expect to tell you for a transaction is to ask them in a usability test or by going to their homes or workplaces and asking them there.
You can also hunt for other examples of forms that deal with similar transactions (as with the examples of job application forms above) and use them as a guide to what users may have learned to expect. This is less satisfactory than asking users, because it can happen that all the forms are doing the same thing—badly.

**Simple Questions May Not Be as Simple as They Look**

In a recent article, “Questioning Questions” (User Experience, 8.2), Jessica Enders points out that answering a question can be more complex than it seems at first glance. In a face-to-face conversation, we can ask extra questions about what to exclude and what to include. When working with a form, that is not so easy. Enders gives the example:

“How long have you been using a walking stick?”

She suggests that we look at this question from the point of view of Maria, who first started using a walking stick when she hurt her hip in 2000. She had surgery in 2002 that gave her relief until 2006. Then the pain returned and she started using the walking stick again. How does Maria answer the question?

Questions that deal with duration and recall of past events can pose significant problems for users. And for any question, the users can make wrong assumptions about what they should include or exclude. You can help your users with these questions by making them more recent and more specific.

More recent: we are more likely to accurately recall events that happened recently. Enders suggests improving the walking stick question by asking:

“In the past four weeks, how often have you used a walking stick?”

More specific: the first or last occasion of doing something is easier to recall than a continuous event. So if the important thing is the overall duration, then you might ask:

“When did you first use a walking stick?”

Perhaps we can all agree on what counts as a walking stick. But the include/exclude problem can be much more challenging. Academic Peter Smith addresses this in a post to his blog, Logic Matters (http://logicmatters .blogspot.com/2009/02/fools-damned-fools-and-designers-of.html), when he describes filling out a form for time off to complete a book:

“Hours per week?” And what on earth does that mean? Official contact hours? Contact hours plus preparation time? Contact hours plus preparation time plus time going to grad seminars that aren’t exactly duties but it would be a Very Bad Thing if no lecturers turned up to? Contact hours plus preparation time plus grad seminar time and background reading to be able to talk informally to various grads? Where do “teaching” hours stop? Who is to say? I press “Help.” Which doesn’t give any clue at all.

Caroline Jarrett (caroline.jarrett@effortmark .co.uk) was a project manager specializing in Optical Character Recognition when she landed a job delivering OCR systems to the UK Inland Revenue for processing tax forms. Caroline runs the usability consulting company Effortmark Ltd. in Leighton Buzzard, UK. She writes a monthly column “Caroline’s Corner” on www.usabilitynews .com. She is co-author of “Forms That Work: Designing Web Forms for Usability” (foreword by Steve Krug).

**Conclusion**

The questions on your form are crucial in establishing what your organization wants from your users. You can help them by:

- Thinking about whether to use a prompt or a fully formed question.
- Providing an explanation of why you need sensitive information and how you are going to use it.
- Guiding the user to answers and help—preferably including an image of where to find it as well as an explanation.
- Explaining why you need the answer, if it is unusual.
- Checking whether your “simple” question gives enough detail about what to include and what to exclude.
Election Results 2009–2010

STC is pleased to announce the results of its 2009 election. Winning candidates are pictured below.

Online voting concluded at 12 noon ET on 9 April 2009. Paper ballots had to be received by that date to be counted as valid votes. Of 8,279 eligible voters, 947 (11.44 percent) cast ballots.

STC’s bylaws, adopted by the membership in August 2007, no longer require the officers in STC’s presidential chain of succession—the incoming president and incoming first vice president—to stand for election. Therefore, incoming president Cynthia C. Currie-Clifford and incoming first vice president Michael Hughes did not appear on the ballot this year.

Congratulations to the winners, and a special thanks to all candidates for their dedication to STC and willingness to devote their time and energy to improving the Society.

**STC Officers**

**President:**
Cynthia C. Currie-Clifford

**First Vice President:**
Michael A. Hughes

**Second Vice President:**
Hillary Hart 617
W.C. Wiese 316

---

Advance your career. Don’t interrupt it.

Mercer’s Online Master of Science in Technical Communication Management

Mercer’s MSTCO program was a high value-adding career step for me, one that I am very glad I took. The curriculum is ... very appropriate to professionals faced with improving organizational performance by communicating ideas, information, and knowledge in an increasingly diverse and dynamic world.

Steve Hoard
2007 Graduate
Senior Division Quality Engineer
Sonoco Products Company
Hartsville, South Carolina

Students may begin the program in August, January, or May.

www.mercer.edu/mstco

---

Mercer’s MSTCO program was a high value-adding career step for me, one that I am very glad I took. The curriculum is ... very appropriate to professionals faced with improving organizational performance by communicating ideas, information, and knowledge in an increasingly diverse and dynamic world.

Steve Hoard
2007 Graduate
Senior Division Quality Engineer
Sonoco Products Company
Hartsville, South Carolina

Students may begin the program in August, January, or May.

www.mercer.edu/mstco
STC Officers (cont’d)
Treasurer:

<table>
<thead>
<tr>
<th>Name</th>
<th>Phone</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aiessa Moyna</td>
<td>535</td>
</tr>
<tr>
<td>Brian Lindgren</td>
<td>379</td>
</tr>
</tbody>
</table>

Nominating Committee:
(two positions)

<table>
<thead>
<tr>
<th>Name</th>
<th>Phone</th>
</tr>
</thead>
<tbody>
<tr>
<td>Suzanne Guess</td>
<td>527</td>
</tr>
<tr>
<td>Sandra (Sandi) Harner</td>
<td>559</td>
</tr>
<tr>
<td>Garret H. Romaine</td>
<td>477</td>
</tr>
</tbody>
</table>

Directors:
(two positions)

<table>
<thead>
<tr>
<th>Name</th>
<th>Phone</th>
</tr>
</thead>
<tbody>
<tr>
<td>Karen Baranich</td>
<td>474</td>
</tr>
<tr>
<td>Lori Corbett</td>
<td>468</td>
</tr>
<tr>
<td>Joaquim Baptista</td>
<td>265</td>
</tr>
<tr>
<td>Makarand (Mak) Pandit</td>
<td>409</td>
</tr>
</tbody>
</table>

The World in Your Words
Celebrating 10 years of on time delivery customer service quality translations online technology

4Translation offers clients any language combination in any format with professional in-country translators, editors, desktop publishing specialists & localization engineers.

4Translation, Inc.
www.4translation.com
info@4translation.com • 1.888.777.9531
Society-level Offices for 2010

By Jackie Damrau
Chair, Nominating Committee

There’s still time to become a candidate for elective office in 2010. The Nominating Committee would like to hear from members interested in learning more about a particular office or in suggesting other members for office.

The STC Board of Directors includes experienced leaders in both business and academe. All candidates should be senior STC members, experienced working at a strategic level, comfortable making decisions collaboratively, and focused on outcomes. They should have Society-level leadership experience as well as professional skills in managing people, projects, and budgets. Being an STC board member is a chance to leverage these skills to lead a not-for-profit organization and to help ensure that the Society continues providing members the same—and better—quality of opportunities and services that you’ve enjoyed.

If you’re interested, complete the candidate nomination form printed here and return it to the Society office or obtain it from the Society office (contact stc@stc.org or call +1 [703] 522-4114 and request that the candidate nomination form be mailed to you). The form has also been posted on the STC website to be filled out and submitted online. You may submit more than one form if you want to recommend more than one person. The deadline for receipt of all forms is 1 July 2009.

The committee invites expressions of interest from all members who would like to help direct and administer Society-level activities. This year, the Nominating Committee seeks prospects for the offices of second vice president, secretary, director, and nominating committee member. In STC’s 2005 election, voters approved a change to the STC Bylaws that makes the director positions “at large”—that is, open to all STC members worldwide.

Members who complete the nomination form (or who are recommended by others) do not automatically become candidates. Once the forms are received, the committee calls all prospects to confirm their interest. Interested prospects are asked to forward additional information, which the committee uses to determine candidacy.

A healthy board combines the talents of seasoned leaders with the fresh perspectives of new ones. Both veteran and senior members are therefore encouraged to complete and submit the nomination form by 1 July 2009.

Candidates for Society Office

Check the appropriate box below and provide the necessary information.

☐ I would like to be considered for the office listed below.
☐ I would like to recommend the person listed in the following section for the office listed below.

Office: ___________

If you would like to be considered for office or would like to recommend another member, please enter the appropriate information below.

Prospective candidate’s name: _________________________
Address: ____________________________________________
Daytime phone: ______________________________________
Email: ______________________________________________

Filling out this form is only a preliminary step in the nomination process. Additional information will be sent to you or the person you recommend. Please remember that when STC members consent to be candidates, they do not automatically become candidates.

Please return a copy of this form by 1 July 2009 to the following address:
Society for Technical Communication, Attn: Nominating Committee,
9401 Lee Highway, Suite 300, Fairfax, VA 22031-1803, Fax: +1 (703) 522-2075

Or email to: stc_election@stc.org

Questions? Contact Diana Buttram at diana.buttram@stc.org, +1 (571) 366-1919.

ERRATA: Please note in the April 2009 issue, we regret the following errors:
• Page 25, for Carolyn Kelley Kinger, read Carolyn Kelley Klinger
• Page 25, for Nominating Committee (three positions), read Nominating Committee (two positions)
The Limitations of Mental Models

By Geoffrey J. S. Hart, Fellow

In the July 2007 installment of this column, I discussed the notion of schemata—what we refer to more colloquially as mental models. The essence of that article was that creating conceptual models to simplify our complex world into something we can deal with more easily offers a powerful tool for understanding. However, as information designers, we must understand the limitations of mental models before we can use them effectively for communication. Psychologist and human factors expert Donald Norman noted several drawbacks of mental models, each of which has implications for design.

Mental models are incomplete.

The world outside our heads is enormously larger than the world within. To hold that entire external reality in our minds simultaneously would require minds as large as the world itself. In consequence, we store only a limited description of that external world. Only the few people who have the proverbial “photographic” memory remember things literally and exactly, but they do so to a more limited extent than most of us think. The rest of us remember only enough detail to reconstruct a reasonably accurate working model. When we speak of user-interface metaphors, we are using a related approach: a metaphor’s power lies in how it uses a schema for something we already understand (something familiar) to let us grasp something less familiar.

When we explain something to our audience by teaching them an appropriate mental model, we must account for the model’s inevitable incompleteness by basing it on only the aspects of the situation most essential for understanding; that is, we must account for our audience’s inability to create and use a truly complete model by emphasizing only those features of a situation that are required for a “good enough” understanding of that external world. By communicating only those key features, we avoid overwhelming our audience with a model that is too complex to grasp and work with.

Mental models have vague boundaries.

Because usable mental models are inherently incomplete, there are limits to how far we should extrapolate from a given model, and it’s often unclear where those limits lie. When we fail to clearly communicate those limits, audiences may apply the model inappropriately. The “trash can” metaphor used by Windows and Macintosh computers seems clear: we put something in the trash when we no longer need it. But that model does not reveal whether the trash empties itself at some unspecified interval, or whether emptying it requires our intervention.

When we invoke a model, we must consider the user’s perceptions of the consequences of acting upon that model. For example, an executive working in an office with a janitorial staff will likely make different assumptions about the “trash can” model than a teenager who allows trash to accumulate to near-disastrous levels. For the trash can, those consequences begin with “what happens after I throw something in the trash?” No mental model can communicate or account for all possible assumptions without becoming impossibly complex. We must understand what we’re explaining sufficiently in order to know how far to take the metaphor, thereby turning the inherent vagueness of the model’s boundaries into clear limits. For the computer trash can, this might mean setting a reminder for the user to empty the trash at least monthly.

Oversimplification leads to unwarranted assumptions.

Even the most skillful information designer cannot always successfully simplify the world’s complexity. The problem with excessive simplicity begins with the vague boundaries we’ve just discussed, but it continues with an inconvenient aspect of human nature: those who use the model will try to take the model in directions we never intended. Because people differ in their perceptions of the most important aspects of a situation, and draw conclusions about a situation based on their preconceptions and biases, we must plan for these inconveniences. There is no easy solution for complexity, but sometimes we can replace a single, large, and complex mental model with a series of smaller and
more manageable submodels. The total number of details may be the same, but by dividing them among several separate submodels, we simultaneously make each submodel more complete (thereby reducing the risk of oversimplification) and less complex.

Another useful trick is to provide thinking aids that make the bounds of what is possible clearer. When we present a blank data-entry field to our audience, the possibilities seem infinite because each user has a different preconception of what they should type. When we supply an affordance, such as the instruction “Type the date here (DD/MM/YY),” we constrain the possibilities, thereby helping our audience make the correct assumptions.

**Mental models are unstable.**

Each of us constantly reasesses our environment, and when we detect changes, we make assumptions about their meaning and rebuild our mental models accordingly. Consequently, even when we build a simple and effective mental model for an audience, we must watch carefully for external factors that will cause them to revise this model. Memorizing details takes effort, so most of us look for ways to permanently remember as little information as possible. In addition, details that we don’t use tend to disappear over time.

Consistency is a powerful tool for stabilizing a mental model. By minimizing changes in what the audience perceives, and only making changes that are truly important, we minimize the number of times our audience will be tempted or forced to modify their mental model. For example, if we were to graph technical writer and information designer salaries from 1980 to 2010, we could use a solid line for the technical writer and a dashed line for the information designer. If we present additional graphs showing other aspects of these professions, such as the mean worker age and duration of their work week during this period, we must never reverse the two line patterns. Each time we do, viewers must build a new model of the meaning of the two lines, and if they fail to do so, interpretation errors will result.

**Mental models also have inertia.**

On the other hand, when we are lazy, fatigued, or distracted (i.e., most of us, most of the time), we may fail to notice changes that require the construction of new mental models. As designers, the knowledge that a changed model will be required reminds us to seek ways to encourage the audience to build that new model. For example, if we now begin graphing the trends for male and female instructional designers, we should choose different patterns—such as a dotted line for both groups, with data points indicated by squares for the men and triangles for the women. The pattern change tells the viewer something has changed, thereby prompting them to learn why and to revise their mental model accordingly. If we used the same patterns that we formerly used for the technical communicators and information designers, viewers may fail to notice the changed context and may interpret the graphs using the old model.

**Mental models can be illogical.**

Microsoft Word used to be famously unstable, so I trained myself to save my work at frequent intervals—typically every 5 to 10 minutes, or whenever I completed something difficult. That seemingly sensible precaution makes less sense now that I have set Word’s autosave feature to do this for me, and since, most days, the software never crashes. But because this mental model costs me almost nothing to act upon, I continue to save files myself. Most experienced computer users develop similarly well-honed intuitions about how their favorite software works, and they act accordingly.

Acknowledging that our audience isn’t always logical, we must recognize the difference between illogical actions that cause no harm, such as my reflex, and those that can potentially cause harm. We should probably ignore the former, because it’s difficult to break long-standing habits and discard the associated models. Although we should, where possible, help our audience to develop more logical and appropriate models, it’s often more important to help them learn to avoid or recover from the results of following illogical models.

**Mental models are parsimonious.**

Most of us build models with the minimum amount of detail required for us to succeed at a given task. When we accept a suboptimal but “good enough” solution, this is known as “satisficing.” Satisficing is a cousin of the scientific principle of parsimony (better known as “Occam’s razor”), in which we strive to avoid overcomplicating an explanation by accounting for even the most unlikely possibilities. The problem with parsimony is that it may lead us to oversimplify our model and ignore less likely possibilities that we shouldn’t ignore. Recognizing that our audience prefers simpler models over complex ones, we must try to provide models that are appropriately parsimonious: they must cover both the most likely problems and the problems that can result from oversimplification. Returning again to the trash can example, parsimony might mean that most users will never worry about what happens to the trashed files until they run out of disk space or suddenly need to recover a file. Only then does the file’s fate become a necessary part of their mental model. A suitably parsimonious model explains both the behavior (the file goes away) and its consequence (it can be recovered from the trash for X days).

**Suggested Readings**


The value of STC’s Technical Communication Summit has been enhanced tenfold!

**NEW THIS YEAR:** The content of more than 100 sessions was captured (audio and visuals) and will be available on the STC website (www.stc.org). Only panel discussions and workshop sessions were excluded. A free demonstration of the technology is available on the website now. SUMMIT@aClick is included in the full registration fee. Attendees will receive a password to access the content at no additional charge. Members who did not attend the Summit may purchase Summit@aClick for $795; however, until 30 September it is being offered at an introductory advance price of $595.

**SUMMIT@aClick**

Did You Miss the Summit?

You can “attend” all the 2009 sessions that interest you at the time and place most convenient and without having to travel.
F.Y.I. lists information about nonprofit ventures only. Please send information to intercom@stc.org. For STC’s complete calendar of events, visit www.stc.org/edu/relatedEvents01.asp.

1. Vienna, Austria
   8–10 June 2009 1.
   The TransAlpine Chapter STC will hold its annual conference and 10th anniversary celebration in Vienna, Austria. For more information, contact:
   Claudia Basso
   vienna2009@stc-transalpine.org
   www.stc-transalpine.org/conferences/vienna-2009/

6. Sydney, Australia
   13–14 June 2009 3.
   The UK and Ireland STC Chapter (UK STC) will hold a two-day event about “Accessibility in Technical Communication and the Workplace.” The event will be held at the Moller Centre in Cambridge (UK). For more information, contact:
   Richard Truscott
   +44 1763 853857
   richard.truscott@btinternet.com
   www.stcuk.org

2. Portland, OR
   The Professional Communication Society (PCS) of the Institute of Electrical and Electronic Engineers (IEEE) will hold its 2009 Professional Communication Conference in Honolulu, HI at the Hilton Hawaii Village. For more information, contact:
   PCS

3. Cambridge, UK
   The Professional Communication Society (PCS) of the Institute of Electrical and Electronic Engineers (IEEE) will hold its 2009 Professional Communication Conference in Honolulu, HI at the Hilton Hawaii Village. For more information, contact:
   PCS

4. Honolulu, HI
   8–12 June 2009 2.
   The Usability Professionals Association (UPA) will hold its 2009 international conference, “Bringing Usability to Life,” in Portland, OR at the Hilton & Executive Tower. For more information, contact:
   Kate Caldwell
   +1 (630) 551-8490
   Chair2009@usabilityprofessionals.org
   www.usabilityprofessionals.org/conference/2009

7. San Antonio, TX
   16–20 August 2009 5.
   The International Society of Logistics (SOLE) presents its 44th Annual International Logistics Conference and Exhibition, themed “The Logistics of Global Security,” to be held at the Omni Mandalay Hotel at Las Colinas in Dallas, TX. For more information, contact:
   SOLE
   +1 (301) 459-8446
   solehq@erols.com
   www.sole.org

8. Dallas, TX
   The Plain Language Association InterNational (PLAIN) will hold its biennial conference, themed “Raising the Standard,” at the Four Points by Sheraton at Darling Harbour in Sydney, Australia. For more information, contact:
   PLAIN
   PLAIN2009@plainenglishfoundation.com

5. Dallas, TX
   The Human Factors and Ergonomics Society (HFES) will hold its 53rd annual meeting at the Grand Hyatt San Antonio in San Antonio, TX. For more information, contact:
   HFES
   +1 (310) 394-1811
   info@hfes.org
   http://www.hfes.org/web/HFESMeetings/09annualmeeting.html

8. Dallas, TX
   22–24 October 2009 8.
   The American Medical Writers Association (AMWA) will hold its annual conference, “Blazing the Trail,” in Dallas, TX. For more information, contact:
   Dane Russo
   +1 (301) 294-5303
   amwa@amwa.org
   www.amwa.org/default.asp?id=433
Touch Screens: From Seemingly Impossible to Increasingly Common

By Ken Schatzke, Senior Member

James Bond movies are showcases for cutting edge devices—from the plausible to the seemingly impossible.

In Quantum of Solace, an MI6 agent shows 007 and M the links between suspects on a digital table simply by touching its surface. Although it seems fantastic, this particular device is much closer to reality than you might think; touch-enabled devices are becoming increasingly common in schools and places of business.

SMART Technologies is a Canadian company on the forefront of touch-enabled devices and related technologies. It produces interactive whiteboards, public displays, tables, and other devices.

The company’s flagship product, the SMART Board interactive whiteboard, is used in schools and businesses around the world. Customers connect the SMART Board interactive whiteboard to a computer and projector, and then see the computer’s display on the board. They can interact with windows, buttons, and other elements by simply touching them on the board’s surface.

I’ve been working as a documentation specialist at SMART since January 2008. In many respects, my work is similar to that of most technical writers—I develop online help, manuals, specifications, and release notes, and work with other writers, editors, usability specialists, and subject matter experts on a daily basis. We have meetings, projects, deadlines, and reviews like any other company.

That said, SMART is an exciting place to work as a technical writer. Our products, along with other touch-enabled devices, represent a significant change in how users interact with the digital world. Educators are using touch-enabled devices, such as the SMART Board interactive whiteboard, to teach students of all learning styles—visual, auditory, and kinesthetic. Businesses are using similar technologies to improve employee communication and collaboration. As an example, employees can now record notes on an electronic whiteboard and then save them to a file for later reference rather than having to write them on a traditional whiteboard with the obligatory “PLO” while searching for a scratchpad or laptop computer to capture them.

Multi-touch technology expands on these possibilities further by allowing multiple users to interact with a single device at the same time. Computing no longer has to be a solitary activity. One of SMART’s newest products, a multi-user interactive table designed for primary school children, is being used by the Spaulding Youth Center in New Hampshire to help autistic students learn and develop social skills.

As the software for touch-enabled devices evolves, interactions in the digital world will begin to resemble those of the real world. Imagine shuffling through the photos on your computer like the ones in your shoebox, or laying out a page on a digital table like tech writers of a certain vintage used to do on drafting tables.

So what does this mean for me and my colleagues at SMART? We’ve needed to expand the traditional software documentation vocabulary to include terms like press or touch (versus click) and show users how to interact with our company’s products. In addition, we’ve had to incorporate more graphical, touch-friendly elements into our help. SMART’s large, diverse customer base requires us to do all of this while considering the needs of audiences ranging from school students to business professionals.

SMART and other companies that produce touch-enabled devices are creating third-party developer communities with the ultimate goal of fostering broad bases of content and applications for their devices. In addition, the next version of the Windows operating system—Windows 7—will include multi-touch functionality, vastly expanding the software ecosystem for touch-enabled devices. New content and application will require documentation. Perhaps you may be one of the technical writers that produce this documentation.

Ken Schatzke (KenSchatzke@smarttech.com) is a senior member of STC and has been working as a technical writer for the past nine years. He lives in Calgary, Canada, and holds a bachelor of applied communications in technical writing from Mount Royal College. He is actively involved with the Alberta Chapter and Lone Writer Special Interest Group within STC.
Create as you want
Publish where you need

Efficiently generate rich technical content that works seamlessly across platforms

Technical communicators who need to print manuals, produce online help systems, update knowledge bases, and craft training modules now have an integrated suite of applications that makes it easy to:

- author standards-compliant content in Adobe FrameMaker® and Adobe RoboHelp®, and repurpose it conveniently by single-sourcing and publishing to multiple channels such as print, HTML, XML, Adobe PDF, and Adobe AIR® runtime
- create interactive user experiences that engage, by adding 3D models and Adobe Captivate® ‘show me’ movies, and letting your images say a thousand words with the powerful capabilities of Adobe Photoshop®

“Technical Communication Suite 2 provides an integrated solution that makes it easy to create multiple deliverables from a single source of content. Instead of jumping from one application to another, my clients can use a consistent and familiar Adobe interface for all of the software tools professional communicators need.”

— Thomas Aldous, President, Integrated Technologies, Inc.