LLYN & BACON SERIES IN TECHNICAL COMMUNICATION
Series Editor: Sam Dragga, Texas Tech University

G SOFTWARE DOCUMENTATION: A TASK-ORIENTED APPROACH
Barker (Texas Tech University)
84 pages ISBN: 0-205-19576-8

LTY TESTING AND RESEARCH
Barnum (Southern Polytechnic State University)
48 pages ISBN: 0-205-31519-4

CONTEXTS: CASE STUDIES IN INTERNATIONAL
CAL COMMUNICATION
Bosley (University of North Carolina at Charlotte)
16 pages ISBN: 0-205-28682-8

IN TECHNICAL COMMUNICATION
browski (University of Central Florida)
58 pages ISBN: 0-205-27462-5

PLES OF WEB DESIGN
kas and Jean Farkas (University of Washington)
75 pages ISBN: 0-205-30291-2

RESENTATIONS FOR TECHNICAL COMMUNICATION
urak (University of Minnesota-Twin Cities)
63 pages ISBN: 0-205-29415-4

ICAL MARKETING COMMUNICATION
Harner and Tom G. Zimmerman (Cedarville University;
l Information Technology Solutions)
84 pages ISBN: 0-205-32444-4

G PROPOSALS: RHETORIC FOR MANAGING CHANGE
hnson-Sheehan (University of New Mexico)
88 pages ISBN: 0-205-32689-7

ICAL WRITING STYLE
(University of Central Florida)
02 pages ISBN: 0-205-19722-1

NING VISUAL LANGUAGE: STRATEGIES FOR
SSIONAL COMMUNICATORS
estelnick and David D. Roberts (Iowa State University)
55 pages ISBN: 0-205-20022-2

ICAL EDITING, Third Edition
ude (Texas Tech University)
20 pages ISBN: 0-205-33536-X

G A PROFESSIONAL LIFE: STORIES OF TECHNICAL
UNICATORS ON AND OFF THE JOB
Savage and Dale L. Sullivan (Illinois State University; Michigan Technological University)
92 pages ISBN: 0-205-32106-2

t our website at
ablongman.com

ISBN 0-205-31519-4
90000
9 780205 315192

BARNUM

USABILITY TESTING AND RESEARCH

Longman Publishers

USABILITY TESTING AND RESEARCH

CAROL M. BARNUM

THE ALLYN & BACON SERIES IN TECHNICAL COMMUNICATION

Bibliography

Craig, James. *Basic Typography: A Design Manual,* Watson-Guptill Publications, New York, 1990.

Horton, William. *Designing and Writing Online Documentation,* Second edition, John Wiley & Sons, New York, 1995.

Horton, William. *The Icon Book: Visual Symbols for Computer Systems and Documentation,* John Wiley & Sons, New York, 1994.

Kostelnick, Charles, and Roberts, David D. *Designing Visual Language: Strategies for Professional Communicators,* Allyn and Bacon, Boston, 1998.

Nielsen, Jakob. *Characteristics of Usability Problems Found by Heuristic Evaluation* <http://www.useit.com/papers>.

Nielsen, Jakob. *How to Conduct a Heuristic Evaluation* <http://www.useit.com/papers>.

Nielsen, Jakob. *Severity Ratings for Usability Problems* <http://www.useit.com/papers>.

Nielsen, Jakob. *Summary of Usability Inspection Methods* <http://www.useit.com/papers>.

Nielsen, Jakob. *Ten Usability Heuristics* <http://www.useit.com/papers>.

Schriver, Karen A. *Dynamics in Document Design,* John Wiley & Sons, New York, 1997.

Technical Communication, Journal of the Society for Technical Communication, February 1998, Volume 45, Number 1, published by the Society for Technical Communication, Arlington, VA.

Technical Communication, Journal of the Society for Technical Communication, May 1998, Volume 45, Number 2, published by the Society for Technical Communication, Arlington, VA.

Technical Communication, Journal of the Society for Technical Communication, November 1997, Volume 44, Number 4, published by the Society for Technical Communication, Arlington, VA.

Wheildon, Colin. *Type & Layout,* Strathmoor Press, Berkeley, 1996.

Source: Reprinted with permission from Ann Dillon and Jeanette Evans.

CHAPTER 3

User and Task Analysis

In Chapter 1, we talked about user-centered design as the process of developing products based on information learned from users. Usability testing is a cornerstone of user-centered design, as it provides essential information about users interacting with products. However, user-centered design does not begin with usability testing. It begins with an understanding of the users and the tasks they perform and the environments in which they perform these tasks. To learn about these critical elements—user, task, environment—you must collect data about users and their behaviors in *their* workplace or home. That means going to the users, not just bringing the users to you. This chapter addresses the methodology of *field studies.* Field studies are not to be confused with field testing, which refers to usability testing in the "field," or the user's workplace, as opposed to the lab (see Chapter 1). Field studies derive from social science and its reliance on *ethnographic* studies, such as the type conducted by Margaret Mead, who lived with people in their community to study their culture. Field studies, in the context of user-centered design, are not as intensive or long-lasting as ethnographic studies, but they can still provide rich data, which can be used as the starting place of user-centered design.

To understand the importance of field studies, we need to first explore the issues of conceptual design and the problems that result when the design of a product does not match the user's concept or experience. This understanding will establish the basis for our discussion of user and task analysis. We conclude by making some distinctions about users and tasks relevant to the World Wide Web.

Conceptual Design

Conceptual design provides the concept, or metaphor, that, if done well, allows the user to intuitively understand the meaning of the interface. No amount of brainstorming or attending meetings will tell developers whether a design concept

will work for users. Nor will surveys or focus groups or any of the other methods for getting feedback from users, as discussed in Chapter 2. Getting the conceptual design right is critical for usability, as usability specialist Jeff Rubin explains: "The *conceptual model* or *metaphor* of a software user interface is the means by which it communicates the software's underlying operations and functionality to a user. It is the highest level of design, of communication, and is one of the prime determinants of the usability and ease of learning a software product" (130). Conceptual design is based on three premises about users and products (Rubinstein and Hersh):

1. Humans always form mental models, maps, or hypotheses about the underlying, invisible processes of a system or machine to help them operate it.
2. A product's conceptual model should match the user's existing mental model or, if a new product is being developed, should make sense to the user, such as the "desktop metaphor" did when Apple introduced the first commercially acceptable computer with a graphical user interface.
3. If the product's conceptual model matches the user's mental model or allows the user to create a mental model for use, then the product will be easier to learn and use.

Problems with Conceptual Models

When is a conceptual model difficult? According to Donald Norman, author of *The Invisible Computer*, difficulty arises with the model "when the controls and actions seem arbitrary, when the system can get itself into peculiar states, peculiar in the sense that the person using it does not know what it is doing, how it got there, or how to recover. When there is a lack of understanding" (174). Although Apple's desktop metaphor has been widely applauded as a usable concept, everyone in the development field knows the conceptual problem created by the Macintosh computer's "trash can," a part of its otherwise successful introduction of a graphical user interface. The concept of the trash can worked extremely well for discarding files that the user no longer wanted, because it matched the user's mental model of placing something in the trash that should be discarded. However, users were troubled and anxious when they had to drag a diskette to the trash can to eject it from the computer for future use, and the lines lit up at Apple's help desk as a result.

To give equal time to Microsoft, a more recent example of a concept that doesn't work is the Start command for Windows 95 and newer products, in which users must go to "start" in order to "stop." See the Abbott and Costello parody in the sidebar entitled "Who's on Start?" for a comical illustration of the problem this really presents to users, as it violates a conceptual model users have. Still another example is the concept of the cellular phone, which is not designed to look and act like a phone but rather like a two-way radio. Even when new users are told that the metaphor for the cellular phone is not a phone, but a radio, they are frequently confused about the requirement to press the "send" button to "receive" an incoming phone call. After all, the cell phone appears to be a phone and you use it like a phone to make and receive calls. So, why doesn't the interface look like a phone, a concept already clearly understood by users?

Who's on Start?

Costello: Hey, Abbott!
Abbott: Yes, Lou?
Costello: I just got my first computer.
Abbott: That's great, Lou. What did you get?
Costello: A Pentium II-22, with 40 Megs of RAM, a 2.1 Gig hard drive, and a 24X CD-ROM.
Abbott: That's terrific, Lou.
Costello: But I don't know what any of it means!
Abbott: You will in time.
Costello: That's exactly why I'm here to see you.
Abbott: Oh?
Costello: I heard that you're a real computer expert.
Abbott: Well, I don't know. . . .
Costello: Yes-sir-ee. You know your stuff. And you're going to train me.
Abbott: Really?
Costello: Uh huh. And I am here for my first lesson.
Abbott: O.K. Lou. What do you want to know?
Costello: I am having no trouble turning it on, but I heard that you should be very careful how you turn it off.
Abbott: That's true.
Costello: So, here I am working on my new computer and I want to turn it off. What do I do?
Abbott: Well, first you press the Start button, and then . . .
Costello: No, I told you I want to turn it off.
Abbott: I know, you press the Start button. . .
Costello: Wait a second. I want to turn it off. I know how to start it. So tell me what to do.
Abbott: I did.
Costello: When?
Abbott: When I told you to press the Start button.
Costello: Why should I press the Start button?
Abbott: To shut off the computer.
Costello: I press Start to stop?
Abbott: Well, Start doesn't actually stop the computer.
Costello: I knew it! So what do I press?
Abbott: Start.
Costello: Start what?
Abbott: Start button.
Costello: Start button to do what?
Abbott: Shut down.
Costello: You don't have to get rude!
Abbott: No, no, no! That's not what I meant.
Costello: Then say what you mean.
Abbott: To shut down the computer, press . . .
Costello: Don't say, "Start"!
Abbott: Then what do you want me to say?
Costello: Look, if I want to turn off the computer, I am willing to press the Stop button, the End button and Cease and Desist button, but no one in their right mind presses the Start to stop.
Abbott: But that's what you do.
Costello: And you probably Go at Stop signs, and Stop at green lights.
Abbott: Don't be ridiculous.
Costello: I'm being ridiculous? Well, I think it's about time we started this conversation.
Abbott: What are you talking about?
Costello: I am starting this conversation right now. Good bye.

Conceptual problems also occur with new Internet users, as the calls to technical support illustrated in Table 3.1 reveal (Kiesler; reported in Nielsen, "Tech Support Tales").

To design an effective conceptual model, which becomes the basis of the interface, developers have to study users at work or at home, observe what they actually do, see what they have created beyond what's available to help them perform

TABLE 3.1 Conceptual Problems for Novice Internet Users

User's Question to Tech Support	What Really Was Wrong	Nielsen's Comments
My email freezes.	The user had never installed the modem (didn't know that it was part of the computer).	Reveals a fundamental flaw in the user's conceptual model of the system.
Modem won't dial.	Someone else was using the telephone.	One more problem caused by a fundamental error in the user's conceptual model of the system: the user would probably not have complained about not being able to use one of the telephones in the house while another member of the household was on the phone, but the user doesn't understand that using the modem is equivalent to making a telephone call. After all, a modem is a *computer* thing.

Source: Adapted from Jakob Nielsen, "Tech Support Tales: Internet Hard to Use for Novice Users" *Alertbox* (1 Apr. 1997) <http://www.useit.com/alertbox9704a.html>.

their tasks, and understand users' goals. In the next section, we discuss the importance of learning about users and their tasks where they work and live.

User and Task Analysis

The goal of user and task analysis is to understand:

- What users' goals are, not just the tasks they perform
- What processes they use to achieve their goals
- What characteristics shape the way they perform tasks and achieve goals (different groups of users may have different characteristics and goals)
- What previous experience shapes users' approaches to tasks
- What is most important to users or what is most helpful to them in performing tasks
- What impact the environment has on their ability to perform tasks

Two examples will illustrate the importance of conducting such user and task analyses, based on the experience of one company that did perform these up-front field studies and one company that did not. In the case of the company that did not perform a field study and therefore did not know the users' environment, the product was a documentation set for banking operations that was intended to be shelved in the workstation or office bookshelf of computer users in a banking center. The product was being developed for a customer based in Hong Kong. The technical communicators who worked on the project assumed that the environment would match that of typical U.S. banking centers. It wasn't until the senior technical communicator went to Hong Kong to initiate the training that she learned that everyone in the banking center worked together in an open room with no offices and no bookshelves. Thus, the document sets were stacked up precariously on top of a printer on a stand in the center of the room, which severely reduced their accessibility.

In the case of a company that did do field studies in anticipation of creating a product that would enter the market after a single competitor had already captured the marketplace, the company found that potential customers would only be interested in switching to the new product if they didn't have to learn new commands and could use the new product like the competitor's product. Still, new users, who hadn't used the competitor's product, wanted a product that was easier to learn than the one dominating the marketplace. With this understanding of the needs of two different user groups, the company developed two ways to use the product: one allowing users to select the competitor's keyboard commands and the other providing a new and simpler path. In addition, the company learned from the users of the competitor's product what they didn't like about the existing product as well as what they did like, which allowed the developers to design better features without sacrificing the features current users liked. The new product provided easier learnability for new users, while maintaining the "old" way for those already comfortable with the other product. The result was that the new product soon captured the major market share, dwarfing and then killing off the competing product.

What You Can Learn from Users

When you make site visits and listen to users, you can learn their vocabulary, observe the tools they use to perform tasks, and discern their mental models for how they perform tasks, frequently from the mistakes they make because the conceptual metaphor of the product they are using doesn't match their mental model. In *User and Task Analysis for Interface Design*, Hackos and Redish outline three broad categories of information that you can learn about users (35):

1. How they define themselves (jobs, tasks, tools, and mental models)
2. How they differ individually (personal, physical, and cultural characteristics, as well as motivation)
3. How they use products over time and the choices they make about the levels of expertise they want or need to achieve (stages of use)

By arranging to see and spend time with different levels of users, you can better appreciate the motivations and needs of novice users, as well as users who are comfortable with a process or product and those who are proficient with it. You can also learn what the motivation might be for users to learn how to use a new product or switch products or processes and how much control they have over such decisions.

Being able to see users of different skill levels is important in developing user-centered products because users' needs change over time, along with their motivation to learn new things or to make advances in their knowledge of a product. Hackos and Redish (79–87) divide any user population into four possible groups, with the characteristics shown in Table 3.2.

Most users will move beyond the novice stage in time, but few will become expert performers. It is critical, however, to understand what novices need as well as

TABLE 3.2 Characteristics of User Populations

Novices	• Fear of failure, fear of the unknown • Focus on accomplishing real work • Impatient with learning concepts rather than performing tasks • Theoretical understanding only—no practical experience
Advanced beginners	• Focus on accomplishing real work • Impatient with learning concepts rather than performing tasks • Randomly access tasks • By adding new and progressively more complicated tasks, begin to develop an empirically based mental model
Competent performers	• Focus on performing more complex tasks that require many coordinated actions • Ability to plan how to perform a complex series of tasks to achieve a goal • Willingness to learn how tasks fit into a consistent mental model of the interface as a whole • Interest in solving simple problems by applying a conceptual framework to diagnose and correct errors
Expert performers	• Focus on developing a comprehensive and consistent mental model of the product functionality and the interface • Ability to understand complex problems and find solutions • Interest in learning about concepts and theories behind a product's design and use • Interest in interacting with other expert users

Source: Adapted from JoAnn T. Hackos and Janice C. Redish, *User and Task Analysis for Interface Design*. New York: Wiley, 1998, pp. 79–87. © 1998 John Wiley & Sons, Inc. (Reprinted by permission of John Wiley & Sons, Inc.)

what competent performers need. For, as Hackos and Redish state, "products will succeed only if they facilitate users having successful first experiences, and only if they also allow for growth and learning and for a variety of patterns of use" (78–79). If developers lose sight of the novice users under the presumption that by now *everyone* must know something about the computer, they will restrict the desire or ability of new users to enter the market for their products. Although product development in the late 1980s and '90s focused on expert user performance to improve the efficiency of use of products in corporate settings, as Nielsen states, the Web has changed all that, putting the focus back on novice users ("Novice vs. Expert Users"). Terry Sullivan, in an online column about the Web, says much the same when he describes a change in the user population from a "more technically-minded and thus perhaps slightly more tolerant" audience to one that is now "increasingly impatient with and less tolerant of 'elaborate,' error-prone, overloaded designs" ("As Simple as Possible"). For an example of the problem in action, see the sidebar entitled "Not His Typing" on pages 92–93 (Laskas).

Stating a slightly different view in his book *The Inmates Are Running the Asylum*, Alan Cooper claims that most users are neither novices nor experts, although all users start as novices. The great middle group he calls the "perpetual intermediates." Although the novices and the experts are a fluid group, the perpetual intermediates tend to remain in this middle category for a long time (182–83). The question is, who are these users?

Since users are the great unknown for many developers, it would behoove developers to get in touch with users directly and learn from them about their needs, wants, and desires. When developers talk to users, observe them at work, ask questions about their goals and objectives, and analyze what they learn, they can begin to get an understanding of important issues that will affect design. Failure to do this can lead to disaster, as Norman explains, using the famous case of Thomas Edison's failed invention of the phonograph. Edison made decisions based on engineering principles and what he felt users would want. For engineering reasons, he decided to use the cylinder for recordings because it was the best technical solution. His competitor chose the disk, it being much easier to store, handle, stack, and label. However, when Edison later gave in to the preferred design, he made his needle track differently from that of the competitor's model, making their records incompatible (think of the Beta and VHS versions of VCRs in the early days). Edison also believed that buyers wouldn't be able to notice any real differences in sound quality between well-known performers and unknowns, so he saved money by making recordings of unknown performers. The customers, nevertheless, wanted to hear their favorite musicians, especially when the recordings of the unknown performers cost about the same as those of the famous ones. Edison failed because he miscalculated the needs, wants, and desires of his users. RCA Victor succeeded, and the rest is history.

For a more recent example of a problem caused by decisions made without learning the wishes of users, take the case of AuctionWatch.com, a site that attracts hundreds of thousands of visitors a month. When it redesigned its site, it faced a user revolt on the Monday the new site launched. As reported in the *San Jose Mercury News*, "Though AuctionWatch had warned users that changes were coming, the company didn't ask for their input or test any of the new additions—a slightly dangerous move for a firm whose business model rests on user loyalty" (Janah). Users complained that the advertising had gotten bigger at the expense of readability, that the new format was harder to use, and that some of their favorite features on the site had been eliminated. The company president said that the company "understands the importance of its users," but perhaps not well enough to understand the need to check with them before making major changes.

The point of these examples and all that we have discussed about users is that it is essential to adopt the following mantra: "Know thy user, for he is not thyself" (Rubinstein and Hersh 8). No one can speak for the user but the user. With this understanding of who the user is *not*, as well as a methodology for learning who the user is, we turn in the next section to a discussion of methods to learn about users' tasks and goals.

What You Can Learn About Users' Tasks and Goals

In addition to learning about the needs, wishes, and desires of users and the environments in which they work, you can also learn about the tasks they perform and the goals they have. Users' tasks are not necessarily users' goals. Tasks are things

Not His Typing

Jeanne Marie Laskas

MY FATHER CALLS. "Can you help me?" he says in an exasperated tone. "I'm trying to order a book for your mother on the computer."

"Of course," I say. "Where are you now?"

He's quiet for a moment. "Where am I?" he says. "I'm on the phone with you. I'm sitting here."

"No, I mean, where are you? On your computer."

"Oh," he says.

My mother picks up the phone. "IS THERE SOMETHING WRONG?" she asks me.

"Please don't shout!" my father says.

"Your father said he could get this book for me," she says. "But he's been at this an hour now and, oh, I don't know, can you help him?"

"Of course," I say. It's interesting to note that she has not asked me to help her, seeing as this is, after all, a book she wants. My father is my mother's link to the Internet. Technology is not her . . . thing. I don't think my father likes being my mother's link to the Internet. This brave new world is, to him, still very new. He feels anything but brave.

My mother hangs up, and my father tells me the history of his problem. He managed to find www.amazon.com. He even found the book. "And I clicked on 'order,'" he explains. "It said I needed to set up an account. I clicked on 'okay.' It asked me for my name and address. I put that in. And then it asked for 'company name.' I don't have a company name."

"Well, you just tab through that," I say.

"Tab through?"

"Yeah."

"What's tab through?"

"Push the tab button on your keyboard. It will skip to the next box."

"Oh. See, they don't tell you that."

No, they don't. Keyboard control is, by now, intuitive to those of us who spend our days with computers. So intuitive that it's hard to conceive of its not being intuitive.

"Well, where are you now?" I ask him.

Pause again.

"Dad, on the computer," I say. My father is an intelligent man. One of those people who skipped a couple of grades. He sailed through medical school. Computer literacy has nothing to do with intelligence, and he is living proof.

"I know that's what you meant," he says. "But I don't know where I am. I pushed 'help' and now the whole bookstore is gone."

"You must have pushed 'help' on your browser," I say. "That's not the help you need."

"I'm supposed to know what kind of help I need? Doesn't that say something is wrong with the help—not me?"

"It does."

"I mean help is . . . help. Or it should be."

"It should," I say. Everything he's saying makes sense. Just sense in a different realm. It's strange to think how two people can speak the same language, but not the same language at all. It's strange to think how this great era of telecommunications, the future that promised to bring people closer together, has put a chasm between my father and me.

And he, at least, owns a computer. He knows how to turn the thing on. He's an entire world ahead of my mother. My mother and I couldn't even begin the conversation my father and I are attempting to have.

"Okay, Dad," I say, "here's what I'm going to do. I'm going to go into amazon.com and I'm going to be you."

"You're going to be me?"

"Yeah, just give me your screen name and your password, and I'll be you, and I'll place the order."

"But ,— "

"It's okay, Dad."

I can tell he feels as though I'm asking him to cheat. My father is a man of integrity. He plays by the rules. He is a person who places a high value on never misrepresenting oneself to anyone, anywhere. How do I explain to him that there is no anyone, no anywhere, not even a oneself in this new land he's tiptoeing through?

My mother picks up the phone. "JOHN," she yells, "WHY DON'T WE JUST GO TO THE BOOKSTORE? I mean, if this is so . . . difficult."

"It's not difficult!" he says. "Please hang up the phone." Then, to me: "Okay, you go in and be me." It's ego pressure, pure and simple.

I click this, click that. "I'm just writing in your address," I say, narrating my every move like a surgeon trying to reassure the patient. When the order is complete, I give him his confirmation number. He thanks me. He says he'll call me again if he gets stuck.

"Oh, but as long as I have you," he says. "Can you just tell me how I double-space when I write a letter?"

"Sure," I say. "What are you using?"

Pause. "What am I using? The computer."

And so I settle in for what is going to be a long conversation, thinking how nice it is to spend some time with my dad.

Source: Originally published in the *Washington Post Magazine*, Oct. 10, 1999, by Jeanne Marie Laskas. Used with permission.

that users do: steps they must take, processes they must complete, acts they must perform—to achieve a goal or objective. Developers sometimes focus on the tasks and lose sight of the users' goals. See the sidebar on page 94, a spoof on the General Motors Helpline, for a humorous example of the confusion that results when the task of "driving" is confused with the goal of "going places" in a car.

Users are primarily interested in attaining their goal; the task is the means to the end. When the task interferes with the users' goal or makes it hard to accomplish the goal, users become frustrated. By observing users performing their tasks and listening to them to learn what their goals are, you can understand ways to match what users want to accomplish, while minimizing those factors that delay or prevent them from reaching their goal.

Cooper divides goals into four basic categories, with examples of each, as shown in Table 3.3 on page 95. He explains that a close parallel exists between corporate goals and personal goals, and both must be satisfied: "Software that fails to achieve either one will fail. Software that fails to achieve the personal goals of its user may not fail at first, but it won't earn loyalty from its customers, and will be very vulnerable to competition that does" ("Goal-Directed Design").

To learn about users' tasks, you need to understand how information or tasks flow for an individual user as well as what happens as the process moves across boundaries from one department to another. You can't get at users' goals merely by asking them, because they may not be able to tell you. Instead, you must first observe users, then talk to them, and match what they tell you with what you see

General Motors Helpline

Helpline: "General Motors Helpline, how can I help you?"
Customer: "Hi, I just bought my first car, and I chose your car because it has automatic transmission, cruise control, power steering, power brakes, and power door locks."
Helpline: "Thank you for buying our car. How can I help you?"
Customer: "How does it work?"
Helpline: "Do you know how to drive?"
Customer: "Do I know how to what?"
Helpline: "Do you know how to drive?"
Customer: "I'm not a technical person! I just want to go places in my car!"

and comprehend. For each task, you will want to gather information from the categories shown in Table 3.4 on page 96 (Faulkner 98).

This method of studying and learning from users about their goals and the tasks they perform to reach their goals not only produces products that satisfy and please users, but also products that contain the features they want and not the features they don't want. Products that fail to satisfy users' personal goals have led to the rise of the "Dummies" books, which generated $121 million in revenue in 1998.

Making the site visit meaningful requires planning and understanding which options you want to use to gather the information you need about users and their goals. In the next section, we look at planning and conducting the site visit.

Planning a Site Visit

When you plan a site visit, you have a number of options to choose from, some of which can be used in combination. These include:

- Shadowing a user for a day, which means following a user like his or her shadow to see where the user goes, what the user does, and how the user performs tasks to accomplish goals.
- Questioning users while they work, which involves watching and talking to users in their own work or home environment. Make notes about where users start and end tasks, what happens next, and whether users accomplish goals. Review these notes with the user to determine if you got it right from the user's point of view.
- Talking after the task, which allows the users to complete a task without interruption and then respond to questions you have about what you observed.
- Think aloud, which is a term you'll see most frequently associated with usability testing, but which can be used on site visits by asking users to speak their thoughts about what they are doing to help you understand what they're thinking as they perform their jobs. This is very tiring for the user, and unnatural, so it is not the most common technique used.
- Critical incident technique, which asks users to explain everything they would do to perform a process or "critical incident." This technique is used when it isn't appropriate to observe the user performing the task, because of the sensitive nature of the task (privacy between doctor and patient, for instance) or because the task isn't performed everyday.
- Scenarios and role play, which provide another means, like the critical incident technique, of seeing what would happen in a particular situation. You might play the part of the customer or the patient, for instance. Critical incident information can be used to create a scenario or role play.
- Cued recall, which requires getting permission to videotape users performing tasks, so that you can review the tape with them afterward and discuss what they were doing or thinking at certain points. This technique can also

TABLE 3.3 The Goal Stack

Goal	Examples
False goals: common in the software industry and easily achieved by programmers' ignoring the user and focusing on the requirements of the code	Save memory Save keystrokes Easy to learn Safeguard data integrity Speed up data entry Increase efficiency of program execution Use cool technology or features Increase graphic beauty Maintain consistency across platforms
Corporate goals: important to the corporation but not necessarily so for the people doing the work	Increase profit Increase market share Defeat our competition Hire more people Offer more products or services Go public
Practical goals: the bridge between the company's goals and the user's goals	Avoid meetings Handle client demands Record the client's order Create a paper model of the business
Personal goals: true for everyone; when users are made to feel stupid, their self-esteem drops and with it productivity drops	Not feel stupid Not make mistakes Get an adequate amount of work done Have fun (or at least not be too bored)

Source: Adapted from Alan Cooper, "Goal-Directed Design." 25 June 1996 <http://www.cooper.com/articles/drdobbs_goal_directed.html>.

TABLE 3.4 Categories of Information to Learn About Tasks

Category of Information	Questions to Answer
The inputs to the task	What information is needed? What are the characteristics of the information sources? What is the availability of the information? What possible errors might occur? Who or what initiates the task?
The outputs from the task	What is the performance criteria? What happens to the output? How does the task performer get feedback about task performance?
The transformations	What is the nature of the decision making? What strategies exist for decision making? What skills are needed? What interruptions are likely to occur and when?
The task composition of the particular job	How often is the task done and when? Does the task depend on any other task? What is normal/abnormal work load? What controls does the task performer have over workload?

Source: Adapted from Christine Faulkner, *The Essence of Human-Computer Interaction.* Englewood Cliffs, NJ: Prentice Hall, 1998, p. 98.

be useful when it's not possible to interview the user immediately after observing the task being performed.

Figure 3.1 is a checklist for planning the materials to take on a site visit.

Asking Questions

Whenever you're using a technique in which you ask questions, you must be careful to ask questions that do not suggest an answer or that do not restrict the response.

Avoid asking "leading" questions, in which you suggest the response you want to get. For instance, leading questions might be, "Did you like the way the software performed?" or "Why didn't you use the online help?"

Instead ask, "What was your opinion about the way the software performed?" or "I noticed that you didn't use the online help. Can you tell me how you solve problems or find information?" By asking neutral questions, you allow the user to answer in his or her own words.

Ask open-ended questions, rather than closed questions. A closed question receives a limited response or a controlled (yes/no) response. Examples of closed questions are: "How many years have you been using this product?" or "Do you

Check	Material
Audio and videotape recording	
	equipment for video recording (perhaps a portable lab that includes camera, scan converter, recorder, microphones, cables, tripod)
	videotapes (check the correct format)
	power strip
	extension cord
	audio recorder
	tapes for audio recorder
	batteries for audio recorder
	extra batteries for microphones
	still camera, if you want photographs of environments, for example NOTE: Always ask permission before you take pictures.
	film for still camera
	extra batteries for still camera
Note taking	
	laptop with cables for plugging it in and extra battery
	diskettes for backing up files
	notebook if taking notes on paper
	pencils, pens
	portable printer with cords and cable if you want to print while on the road
	paper for the portable printer
Papers and other materials for working with users	
	folder for each participant
	copy of correspondence that went to that site
	release form for each participant at site (and extras)
	information from recruiting or screening questionnaire
	user profile questionnaires (take extras in case you talk to other users)
	other lists, questionnaires, forms, scenarios, or props for planned activities
	supplies for doing a group activity to capture a large process flow (for workflow analysis): poster paper, colored markers, color sticky notes (if you are going to use this technique)

Figure 3.1 Checklist for planning materials for a site visit
(From the work of Janice Redish, Redish and Associates, Inc., <http://www.stcsig.org/usability/resources>, or redish@ari.net.)

use the spreadsheet application of this product?" Open-ended questions allow the user to share information more broadly and freely. By asking open-ended questions, you frequently learn a lot more than you anticipated. Examples of open-ended questions include: "How would you describe your experience level using this product?" or "Which applications of this product do you use?"

Illustrations from Site Visits

Site visits can teach you a lot about what's important to users so that you don't eliminate something they need and want, while you're contemplating "improving" or upgrading a product. Several examples from site visits illustrate the value of learning what's important to users. For instance, what if your company is thinking of doing away with print documentation in favor of online documentation? How do you know whether this is a good decision? A site visit can show you. In one case, a company had two distinct groups of users of the documentation: system administrators and programmers. Documentation was a combination of print and online information. However, when these two groups were studied in the context of their work, the company learned that when the system administrators were troubleshooting problems, they would spread out half a dozen different books on their desk so that they were sure they were taking into account everything they needed to. Online help would have made it much more difficult for them to do the same thing. On the other hand, the programmers typically had a specific question about a particular function, which could be answered most quickly with the online help and which then enabled the programmers to get quickly back to making the changes in the code ("Making Online Information Usable").

In another situation, the design team was working on a network diagnostic system for a nationwide communications company. The team assumed that a nationwide map showing the network connections would be the appropriate display for the network. When they made their site visit, they learned that the company organizes the network according to customer accounts, not geographic location, because their first priority was to know who was experiencing a problem so that they could contact the customer, rather than where the problem was occurring (Wixon and Comstock). Of course, unexpected surprises can change your plans, as the sidebar by Robi Gunn, entitled "Field Trials: Trials and Tribulations of a Field Visit" shows (pages 100–101). However, even in those situations where everything does not go as planned, you can learn a lot.

So far, we've discussed ways to get information about users in the context of their workspace. A growing market is information appliances in the home. Some companies now make it a priority to conduct site visits in people's homes to learn about this environment and the user's needs. When they fail to learn what users want from information appliances, they may end up with products users may desire but can't use, as Dave Barry so aptly describes:

> . . . Here is what really concerns me about these new "smart" appliances. Even if we like the features, we won't be able to use them. We can't use the appliance features we have *now*. I have a feature-packed telephone with 43 buttons, at least 20 of which I am afraid to touch. This phone probably can communicate with the dead, but I don't know how to operate it, just as I don't know how to operate my TV, which has features out the wazooty and requires *three* remote controls. . . . And now the appliance manufacturers want to give us even *more* features. (D8)

If it isn't possible to see users in their home, a few techniques for getting feedback from them can add a lot to your understanding of their needs. You can set up phone surveys in advance and plan a task with the user, then phone the user and ask the user to tell you what he or she is doing while performing the task. With software or hardware that is already in use, this approach gives you information about the features people use and don't use, as well as what they like or dislike about the product. If the product is software, the user can send screen captures ahead to discuss problems, or you and the user can have the software open together so that you can see what the user is talking about.

Point-of-sale research is also an excellent source of information about consumer products. If you are planning to develop a product that will compete with others already in the marketplace, you can go to the place where shoppers would buy such a product and watch what they do. Using contextual inquiry, you can interview people at the point of a purchasing decision to find out what motivated them to buy a particular product. What did they find compelling? What expectations do they have for the product? For its documentation? If the product is available for practice, what did they try out?

Constraints on Doing Site Visits

It is easy to see the advantages of doing site visits. So, why aren't they a routine part of the development process? Well, for one, there is cost involved: the cost of traveling to the site, plus lodging and food while there, the cost of lost productivity at work while you're at the site, not to mention the cost of analyzing and presenting the mounds of data you'll take back to the office following a site visit. One consulting group estimates the time required to perform this analysis as four hours of analysis time for every one hour on site ("Contextual Enquiry"). Another reason that some companies don't do site visits is that designers frequently don't give sufficient credence to their value, thinking they already know about the users from other sources like marketing, surveys, and technical support, or worse, thinking that the users are like themselves.

In addition, many companies are slow to realize the full spectrum of activities that encompass user-centered design. Ehrlich and Rohn (76–78) describe four stages of acceptance of user-centered design: from skepticism to partnership (see Table 3.5 on page 102). Some companies are still stuck in Stage 1. Even companies that are in Stage 3 may include usability testing as part of their standard development process, but they have not yet attained Stage 4, where they understand the importance of involving customers or users before development begins.

Even with companies that have reached Stage 4, certain constraints have to be recognized when going into the field to do research about users and tasks:

- *Time constraints.* Users or customers can spare only a limited amount of time, and your company can spare only a limited amount of your time away from the office. Careful planning is critical to make a site visit a success, as time is of the essence.
- *Budget constraints.* Even with funding for site visits, the number of observations and interviews will be limited by both time and budget. Again, planning is the key to maximizing the opportunity.

Field Trials: Trials and Tribulations of a Field Visit

Robi Gunn

Both Feet First

I was excited. After two years of documenting an application, I was finally flying out to meet some real live users and observe how they used our product, and if I was lucky, how they used our documentation.

I dutifully and eagerly prepared myself for the visit. I read books and STC articles on field visits and questionnaires. I was on a quest, and dangerously close to realizing a dream. At last, I would be able to define my audience, and gage the usability of the online help and hard copy manual. I would finally get the answers to my questions directly from a group of users.

My goal has always been to write meaningful documentation, that is, meaningful from the users' perspective. At work I had asked questions and tried to determine who that elusive average user is, and what information the average user likely requires. I never got a sense that this had been previously defined, or that anyone could really give me that vital information. So I did what most of us do; I guessed. I tried to imagine what the average user was like, and then write for that person. Now that I was going to meet some users, they could confirm that my guesses had been correct, or provide me with the information I needed to refocus the documentation to fit their needs. This was the first and most important piece of the puzzle for me. From there, it would be smooth sailing with just a few adjustments and fine tuning (I hoped). I couldn't wait to hear what they thought of the documentation.

Ground Zero

"What do you mean you didn't know that there was documentation?" I said it calmly and politely, but that little voice inside my head was shrieking. How in the world had these people used our product for two years and not been aware that there was documentation?

Besides providing hard copy manuals and online help, we had just recently placed all our documentation on a web site and created a CD. It was a shock to me that they could have gone two years without realizing that there was documentation. More mysterious yet, they had documentation at their fingertips, literally one keystroke away. I would have thought that at some point during the two years, someone would have randomly (accidentally?) pressed the F1 key. In two years, no one was ever curious enough to wonder what that help menu was, or what that help icon on the screen might do if you clicked on it?

It felt like a bomb had been dropped. I had flown out to meet the users, observe them using the product, and interview them about the documentation to establish a baseline. I was acutely aware that I was now standing at ground zero. I put away my carefully prepared folder of questions and research. It no longer had any meaning. I quickly came to the conclusion that I would have to go backwards before I could go forward.

Plan B

So instead, I began to demonstrate the online help. Prior to the visit, an agenda had been established that included meeting with various departments. During each meeting I gave a presentation of the basic features of the online help: Contents page, Index, Full Text Search feature, and context sensitivity. I indicated what procedures were documented and how they would typically (still just my best guess) use the online help to answer their questions. I got a great response. Most of the users seemed happy to find out that there was documentation, and the initial reaction was that it was "pretty good stuff." I also got the names and phone numbers of users that I could contact at a future date, and I promised to send copies of the manual and CD. I also gave them the URL for our web site, but found that there was some question if they had access to it because of security on their firewall. I thanked them for their time and interest, and headed back home.

Meanwhile, back at the ranch . . .

On the flight back, I mulled over how I would present this information to my boss. I concluded that it was a step in the right direction to find out the truth, regardless of how painful that truth turned out to be. I had established a baseline; it was just several hundred feet farther back than I had previously assumed. I had re-learned that valuable lesson: never assume anything. I also concluded that you have to start somewhere, and at least now we knew where that starting point was. A very important benefit from the trip: I had established contacts for the future. We could continue our mission to provide documentation that the users wanted.

Back at home, I sent copies of the manuals and CDs. I wrote thank you notes to our hosts. Then I began to follow up.

I verified how manuals are ordered and shipped. I confirmed that we had our processes in place, and that they were being followed. No problems there.

I collaborated with our Training Department. When the class for this product was taught, I made arrangements to provide pizza during a working lunch so that I had an opportunity to introduce and demonstrate the online help. Online help is now a regular part of the course. At the end of class for any product, all students now receive a copy of the documentation CD. In addition, the Publications Department is now registering documentation to get a handle on "who" has "what" documentation and we have started sending questionnaires.

I am planning to follow up with the contacts that I made now that they have had a chance to use the documentation. I will probably retool the questions that I had developed for the original visit into a questionnaire to send to them. A repeat trip to follow up and show our commitment to our users wouldn't be a bad idea either. What is more flattering to a user, or demonstrates more interest, than to seek input from them? And with that input, you really can write meaningful documentation, which is all I ever wanted to do.

Well, it's about six months since I made my field visit. I'm not where I want to be, and I'm not where I expected that I'd be, but I am hopeful about the future. I have an old plaque that says: "The longest journey starts with the first step." I have begun.

Source: Reprinted from *Usability Interface* (Oct. 1998). <http://www.stcsig.org/usability>.

- *Political and ethical constraints.* It's difficult to be in someone's workspace or home for very long, and you will never be invisible, so your presence puts a strain on the companies or families that agree to host you and on the individuals you'll be observing. You may also be observing sensitive information that will require your being discreet. You must also get permission to take photographs, set up video cameras, or audiotape, and you must explain what you will do with the information you gather and record.

Although these constraints limit some direct access to users in their environments, an increasing number of companies that adopt a user-centered design

TABLE 3.5 Four States of Acceptance of User-Centered Design

Stage 1: Skepticism	Typifies organizations that have never been involved in user-centered design (UCD). They fear such processes will delay product development; they tend to focus on product features and schedule deadlines.
Stage 2: Curiosity	Companies recognize that their products need help and they become curious about what UCD can offer.
Stage 3: Acceptance	UCD people are on the development team.
Stage 4: Partnership	The organization has a high-level commitment to UCD, which includes getting customers involved early in the process.

Source: Adapted from Kate Ehrlich and Janice Anne Rohn, "Cost-Justification Usability of Engineering: A Vendor's Perspective," *Cost-Justifying Usability*. Eds. Randolph G. Bias and Deborah J. Mayhew. Boston: Academic Press, 1994, pp. 76–78.

process recognize the need for design team members to go on site. If you work for one of these companies, you may be overwhelmed with all the data you gather from a site visit and wonder what to do with it. In the next section, we examine your options for using the information.

What Happens After a Site Visit

When you return from a site visit, you will be armed with a rich load of information that has to be digested and shared with other members of the development group. If you received permission to record your site visits using audiotape, you can create a transcript of your interviews from the audiotapes. If you received permission to videotape, you can edit the videotapes into a highlights tape to show the development team the key issues discovered from the site visit. If you have taken photographs, they can be a useful reminder of the environment in which users work. In addition, you may have received permission to take *artifacts*, the objects that users create or assemble to help them perform their tasks. Artifacts may include sticky notes that users post as reminders to help them work with an existing product, self-created templates or quick reference cards, or books like the "Dummies" titles that users rely on when the documentation isn't clear or isn't available to them.

You will probably write a report to share your findings with the rest of the development team. In the report, you will want to include:

- User analysis, which leads to the creation of user profiles.
- Task analysis, which leads to task lists.
- Environment analysis, which leads to the development of constraints on how users work (such as noise levels, visibility, space requirements, interruptions, etc.).
- Methodology used for data-gathering.
- Recommendations for further research, which often includes a survey to get a larger response regarding issues learned from the site visit.

This report, along with the artifacts you bring back from the field, and the photographs, audio transcripts and video highlights, will prepare the team to begin the process of product development.

Cooper recommends using this information to create *personas* that bring users to life and thus help developers design products that truly match the goals of different levels of users. These personas do not represent the generic "user," which Cooper claims is so vague and loosely defined that it gives developers license to keep adding features and making decisions that the supposed user would want. Rather, Cooper's personas are intended to speak for specific users, each with a name, a car, a family, a job, a residence, and a life. To bring them to life, he purchases faces for each user from stock photo files or from the Web. According to Cooper, "Personas are the single most powerful design tool that we use" (*Inmates* 130). At every meeting, each member of the design team, as well as the client, is handed a sheet with the cast of these characters. All questions are posed in light of what Mary or Bill or Julio or Nobuku would want, not what the "user" wants. Some companies using this concept take it a step farther and make posters of their users, which they put up on the walls of the meeting room. One company makes t-shirts of the personas to be worn by different members of the development team. Others put up quotes received from users during site visits with pictures of the users working, as well as flowcharts of work processes.

In addition to creating personas, Cooper recommends creating *scenarios* to focus on the tasks users perform as they work to accomplish their goals. These scenarios are based on the personas, which are generated after studying users in the field. Scenarios are of two types:

- *Daily-use scenarios*, which reflect the primary actions users will perform most often.
- *Necessary-use scenarios*, which include all the actions that must be performed, although typically infrequently.

Because daily-use scenarios reflect uses that must be learned by all, they must be simple and intuitive for new users and they must also allow shortcuts and customization once users become more proficient. Necessary-use scenarios may be more numerous but they won't need customization, as they are used infrequently. Scenarios and personas provide a means to personalize both users and usage, based on what is learned from site visits. In this way, the knowledge from site visits isn't lost or generalized, but stays focused on developing products that match users doing real tasks to attain real goals.

Knowing about the users and the tasks they perform is the critical first step, but before we leave this subject, we must also establish a basis for product development that reflects the ways in which users learn. In the next section, we take up the subject of users' learning, so as to be able to apply this information in creating products that match users' learning styles as well as their goals.

How Users Learn

In addition to what you can learn from users by studying them in their environments, there is much that can be applied from research on how users learn new products. This research derives largely from the field of cognitive psychology, the

science that studies the way people perceive and remember things, how they store information, and how they organize and retrieve that information when they need it. In this section, we look at some of the aspects of short- and long-term memory, followed by a discussion of learning styles, the specific needs of adult learners, and the research on minimalism and adult learners.

Short- and Long-Term Memory

As learners and users of information, we store information in either short-term memory for immediate use or long-term memory for later or continual use. A well-known outcome of cognitive research is the rule of 7, plus or minus 2, which holds that people can retain seven pieces of information in short-term memory, plus or minus two pieces (Miller). Thus, we can remember 5 to 9 numbers, 5 to 9 items in a list, 5 to 9 steps to perform a task, and so forth. When the U.S. Postal Service attempted to change zip codes from 5 digits to zip + 4, they reached the maximum number that the brain can store readily in short-term memory for immediate recall. Even at this upper limit, it is hard for many to remember the extra four numbers, so they don't use them. The same problem occurs in some large metropolitan areas, where users have to remember different areas codes for phone numbers in the same city. This requires users to dial 10 numbers to make a call, thereby exceeding the normal retention rate for short-term memory.

For long-term memory, research shows that we develop a *schema*, or pattern of action and behavior, that allows us to plug new information into one of our already familiar patterns or to modify the pattern as needed to fit the new information. When we find ourselves faced with a pattern that does not match one of our existing schema, we have to create a new mental model. With a new experience, however, we use as much as we can from what we already know to help us learn something new. When reading or hearing new information, one very important tool we use to fit the new information into a pre-existing mental model is vocabulary. Hearing or seeing words for which we know the meaning helps us use these to unlock new meaning. Of course, problems arise when familiar words are used in new or different ways or when words are misinterpreted because the user's schema does not match the new situation. The following humorous example points up the problem:

> A man and a woman walk into a bar and order a drink for every person in the bar. They are very happy. When the bartender asks them why they are so happy, they reply:
>
> "We finished a jigsaw puzzle in only two months."
>
> "Two months?" the bartender exclaimed, "it's not supposed to take that long."
>
> "That's not true," said the woman, "It said 2 to 4 years on the box."

For more in the same vein, see the sidebar entitled "Communication Gap."

Turning to real-world examples, take the case of the graduate student who was not a seasoned Web user, but who was required to study a Web site and report on its usability. He chose the Sony Web site, because he was interested in purchasing a camcorder. He hadn't purchased Sony products before and didn't know much about the features of its camcorders or any camcorder, for that matter, so he began his search by clicking on "My First Sony Products," reasoning that this would be the place to go for someone who hadn't bought Sony products before. Much to his surprise, but of no surprise to his classmates with children, he found himself at a page of children's products. Clearly, his mental model (and corresponding vocabulary) did not support Sony's model, which used its own marketing metaphor—"My First Sony"—on the assumption that visitors to the site would already be familiar with its meaning. For more examples of problems with vocabulary, see the sidebar on page 106 entitled "Computer Illiteracy," which has been widely circulated on the Internet (Carlton). Although some of these examples may be dated, they are still representative of the problems users typically experience when their schema does not match that of the product.

Experience affects how easily we may want to create new schema, based on our memory of previous pleasure or pain with similar situations. For instance, if we have a negative experience with learning a particular software product or trying to use a particular manual or online help system, we are likely to feel negative about learning how to use another similar product or going back to the manual or online help for future support. Thus, emotions play a significant role in our desire to learn about new products or have new experiences. Motivation also plays an important role in our enthusiasm for learning and our willingness to struggle through complex processes to gain knowledge of a tool or acquire a skill.

When it comes to readers of documentation, whether in print or online, we know from studies that readers:

- Use documents as tools
- Decide how much attention to pay to a document

Communication Gap

A judge was interviewing a woman regarding her pending divorce, and asked, "What are the grounds for the divorce?"

She replied, "About four acres and a nice little house in the middle of the property with a stream running by."

"No," he said, "I mean what is the foundation of this case?"

"It is made of concrete, with brick and mortar," she responded.

"I mean," he continued, "What are your relations like?"

"I have an aunt and an uncle living here in town, and so do my husband's parents."

He said, "Do you have a real grudge?"

"No," she replied, "We have a two-car carport and have never really needed one."

"Please," he tried again, "is there any infidelity in your marriage?"

"Yes, both my son and daughter have stereo sets. We don't necessarily like the music, but the answer to your question is yes."

"Ma'am, does your husband ever beat you up?"

"Yes," she responded, "about twice a week he gets up earlier than I do."

Finally, in frustration, the judge asked, "Lady, why do you want a divorce?"

"Oh, I don't want a divorce," she replied. "I've never wanted a divorce. My husband does. He said he can't communicate with me."

Computer Illiteracy

- An exasperated caller to Dell Computer Tech Support couldn't get her new Dell computer turned on. After ensuring the computer was plugged in, the technician asked her what happened when she pushed the power button. Her response, "I've pushed and pushed on this foot pedal and nothing happens." The "foot pedal" turned out to be the computer's mouse.
- Another customer called Compaq Tech Support to say that her brand-new computer wouldn't work. She said she unpacked the unit, plugged it in, and sat there for 20 minutes waiting for something to happen. When asked what happened when she pressed the power switch, she asked, "What power switch?"
- Compaq is considering changing the command "Press Any Key" to "Press Return Key" because of the flood of calls asking where the "Any" key is.
- An AST customer was asked to send in a copy of her defective diskettes. A few days later a letter arrived from the customer, along with photocopies of the floppies.
- A Dell customer called to complain that his keyboard no longer worked. He cleaned it by filling up his tub with soap and water and soaking the keyboard for a day, then removing all the keys and washing them individually.
- A Dell technician received a call from a customer who was enraged because his computer had told him he was "bad and invalid." The technician explained that the computer's "bad command" and "invalid" response shouldn't be taken personally.

Source: Adapted from Jim Carlton, "Computers: Befuddled PC Users Flood Help Lines, and No Question Seems to Be Too Basic." *The Wall Street Journal*, 1 Mar. 1994, p. B1.

- Jump into documents (there is no shared "starting point") even when documents are labeled "read me first"
- Need to find information easily when they want it (navigation is critical to their success)
- Formulate a question
- Skim, skip, and read only as far as they think they must to get the answer to their question (even stopping mid-sentence)

On this last point, I have observed participants in usability tests of documentation reading only part of a sentence and then acting. The result is that the context of an additional chunk of information, which follows a command, is missed by the user. Because users will read only as far as they think they need to, they will tend to stop reading when they receive an instruction or command before they know what the result will be.

Writing documentation that is structured in this fashion is a violation of the *given-new* contract (Haviland and Clark; reported in Redish "Understanding Readers" 31). The given-new contract is the expectation people have that new information will be presented in a framework that is already known or has previously been given. At the sentence level, it means that readers get the contextual or known information first, followed by the new or resulting action. Using this pattern, the user can decide whether he or she wants to perform the step before doing so. One research study found that when readers get the new information before the given or contextual information, they can choose to do one of two things:

- Put the new information in a "buffer" until they get the context of use to understand it
- Guess at the context and act without waiting

The readers in the study jumped the gun and acted (Dixon 1987; reported in Redish "Understanding Readers" 33). From my observations of usability tests of the documentation, I can confirm that readers jump the gun frequently, as the following examples show:

- After apparently reading only the first five words of the instruction for inserting the diskette, the user said, "You don't tell me where."
- After apparently reading only the first six words of an instruction asking the user to click yes or no, the user clicked "no" and said, "Nothing's happening here. You need to tell me how long this is gonna take."
- After apparently reading the first seven words of an instruction to conclude a task, the user asked, "How do I know if I'm done?"

In all three cases, the information the user sought followed the command, but the users stopped reading and jumped the gun as soon as they identified an action they could take.

In screen design, the given-new contract applies to a consistent design, so that users will see new screens in the context of a familiar pattern from previous screens. For instance, boxes that contain choices should be presented in the same order on every screen. If users are accustomed to clicking in the lower left to go back and the lower right to go forward, they will be confused if the order changes or a different element is introduced. As well, screen design should match users' normal task flow. If tasks move from left to right, the screen should reflect that process. The Next or OK button should be on the far right as the logical place where the user ends up. If the task moves from top to bottom, the Next or OK button should be at the bottom of the screen.

In addition to these memory and consistency issues and the impact they have on learning, we must also consider the different ways in which people learn, a subject we explore in the next section.

Learning Styles

People are different, and so are the ways in which they learn. Learning styles can be characterized in the following four ways:

1. Doing
2. Imagining
3. Reasoning
4. Theorizing

The characteristics of each of these learning styles are presented in Table 3.6 (Coe 57).

TABLE 3.6 Characteristics of the Four Learning Styles

Doing	Imagining	Reasoning	Theorizing
Relies on experiments and plans	Relies on imagination	Relies on deductive reasoning	Relies on theoretical models and inductive reasoning
Enjoys new experiences	Views experiences from multiple perspectives	Prefers hypothetical experiences	Tries to integrate disparate experiences
Takes risks	Brainstorms before acting	Acts in narrow, prescribed manner	Thinks of risks at an abstract level
Adapts to circumstances	Internalizes circumstances	Does not adapt well to changes in circumstances	Raises circumstances to theoretical level
Uses trial and error for problem solving	Relies on insight for problem solving	Uses hypotheses for problem solving	Relies on syllogistic reasoning for problem solving
Is at ease with people	Is people-oriented	Is not at ease with people	Is at ease with people on a theoretical level
Is impatient	Likes to counsel people	Has narrow technical interests	More concerned with sound logic than facts
Excels in marketing or sales	Excels in human resources and counseling	Excels in engineering	Excels in research and development

Source: Mariana Coe, *Human Factors for Technical Communicators*, 1996, p. 57. © 1996 John Wiley & Sons, Inc. (Reprinted by permission of John Wiley & Sons, Inc.)

Which learning style is yours? Which is your user's style? Obviously, one style does not fit all. Therefore, you must provide various ways for people to learn. Some want to learn by a tutorial or guided tour, some (although few) read the documentation first as a method of learning, some want to explore (using help when they need it), others like wizards (which present dialog boxes that ask questions or allow users to select options to perform a process). Still others want to be left alone, trusting that the interface will be understandable because it will match a mental model of previous action or will make sense intuitively.

When users skip the tutorial or don't read the manual, they are typically motivated by "the paradox of the active user" (based on research by Carroll and Rosson; reported in Nielsen, "Paradox"), which means that they are driven to be productive, to learn by doing, not by reading. Their goal is "throughput," the outcome, even if the method they choose to learn the product is less efficient than going through the manual or using the tutorial. Thus, the paradox. It does no good to instruct users to read the manual, as Figure 3.2, which represents a sticker on a VCR, does. Users make their own choices, based on their learning style.

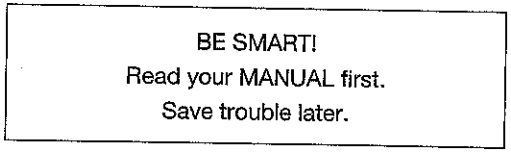

Figure 3.2 Sticker placed on new VCR

If help is available, active users rely on it, but the time spent getting to the right information when they need it is "downtime," not productive time, since it doesn't contribute to the completion of the task or the accomplishment of the goal. Because we know this about active users, we can decrease the downtime expended when we know the words they will use to search a help file to find the answers they seek. If the help topics are categorized by the features of the tool and not by the tasks that users want to accomplish, help will not be helpful, because the terminology of the tool is not known to the user. For instance, if the user wants to write an email message, "compose" (the word used by MSN's Hotmail) may not register as the place to go for this task.

Objects, particularly on-screen objects that are part of a graphical user interface (GUI), communicate to users by providing one or more of the following four "clues" about their use (Coe 167–68):

- *Affordances.* The actual and perceived properties of an object that suggest how we should interact with it. A chair's affordance is its "sitability."
- *Constraints.* The properties of an object that limit what we can do with it. We can stand in a chair to reach the top of the refrigerator (nothing in its design prevents this), but we can't use it to boil water (the design does not accommodate this objective).
- *Mappings.* The properties that suggest how we should interact with or use it. A door with a plate on it (and no handle) suggests that you push it open.
- *Visibility.* The degree to which the object conveys its affordances, constraints, and mappings. If the door has a handle, rather than a plate, can you tell how you should open the door? Does it want to be pushed away from you or pulled toward you? The degree of certainty the object conveys is its degree of visibility.

Donald Norman's classic book on this subject, The *Design of Everyday Things* (formerly *The Psychology of Everyday Things*) explains the problems that arise when a design does not match the user's mental model, using such common examples as doors that do not show how they should be opened, knobs on stoves that don't clearly suggest which knob turns on a particular burner, and faucets that don't suggest which way to turn the handle to get water. When objects communicate logically, intuitively, and consistently, users experience a high degree of success with the match to their mental models. When they do not, users are frustrated, frequently blaming themselves for failing to understand.

As the door and the stove knob problems illustrate, not all objects have good "visibility." However, objects should not be thought of as merely material. The characteristics of objects apply equally well to computer objects, or metaphorical concepts, such as the trash can object in the Macintosh GUI, which didn't suggest to users that it should be used to eject a diskette. Another object or concept that causes problems for users is *drag and drop*. A learnability issue is connected to the drag-and-drop concept whenever it is not transparent to users that this action is required to perform some tasks. Users must learn four concepts to use drag and drop ("Drag and Drop Has a Learning Problem"3):

1. What objects can I drag?
2. Where can I drop them?
3. What's it going to do when I let go?
4. If I don't like it, how do I undo it?

The first two problems relate to a lack of obvious affordances. Drag and drop isn't readily visible. It has to be "learned." The third problem results from a lack of consistency with the drag-and-drop feature. In many products, drag-and-drop will copy an object in one instance but delete it and replace it elsewhere in another instance. The user cannot intuitively determine which result will occur. Question 4 is related to question 3, in that if the user performs an action that doesn't achieve the desired result, the user needs to be able to undo it before it's too late. The undo feature may not serve as a real option, as the user may have already gone a step beyond the ability to undo some action. When this is a possibility, the system needs to provide verification questions to confirm the user's choice.

Adult Learners

Although learners of all ages may prefer different learning styles, adult learners have additional issues that must be given special consideration when designing products for them. These are largely based on adults' motivation for learning vs. that of children.

One theory of adult learning, called *andragogy*, emphasizes that adults are self-directed and expect to take responsibility for their actions. Andragogy presents the following conditions for adult learners (Knowles 55–61):

- Adults have a deep need to be self-directing.
- Adults need to know why they need to know something.
- Adults need to learn experientially.
- Adults need to approach learning as problem solving.
- Adults learn best when the topic is of immediate value.
- Adults enter into a learning situation with a task-centered orientation.
- Adults are motivated to learn by both external and internal stimuli.
- Because of life experiences (both pleasant and unpleasant), adults develop habits and biases that shape their approach to learning.

A survey of business professionals who were experienced computer users found that four of the six highest-rated usability characteristics (out of 21 choices)

related to exploratory learning, which is the preferred mode of adult learners (Nielsen, "What Do Users Really Want?"). These findings support the research of John Carroll and his colleagues, which led them to prescribe a documentation approach called *minimalism*, a subject we address in the next section.

Minimalism for Adult Learners

As we presented in Chapter 2 in our discussion of heuristics, John Carroll and his colleagues at the Watson Research Center at IBM studied the ways in which adults learn new software, focusing their research on the documentation, particularly the tutorial, as a method of learning. The research began by watching people struggle with documentation and the task of learning how to operate a system or software tool. The description of the struggles of adult learners matches the issues adult learners experience when information is not presented in the way they want to learn it:

> Our interpretation of our subjects' struggles was that they were actually making rather systematic attempts to think and reason, to engage their prior knowledge and skill, to get something meaningful accomplished. They did not seem to be getting appropriate guidance and feedback from the systems and documentation they were using, even though they were being presented with a huge amount of information through these channels. For example, although they . . . made a great variety and number of errors, their materials did not support error recognition, diagnosis, or recovery, and the systems did not provide general undo functions. (Mack, Lewis, and Carroll; reported in Carroll, "Reconstructing" 2–3)

Based on these findings, Carroll and his colleagues pursued a different approach to the design of documentation, one that would give readers what they want in the way they want it. The approach, called minimalism, derives from two main principles of cognitive psychology:

1. Users construct their own mental models based on schema.
2. Users want to be actively involved in learning right away.

Even when users are learning a product for the first time, they still want to get started right away with tasks. A small group of users, perhaps 15% (Penrose and Seiford; reported in van der Meij and Carroll 42–43), will read a manual cover to cover. A second group begins at the beginning but then abandons the manual, returning to it only to look for some specific information. A third group uses the manual "as a last resort." The manual must support the needs of all these groups, and for all it must not look intimidating, which means it should take a minimalist approach. This concept gave rise to minimalist manuals, which have often been misunderstood and misinterpreted to mean brevity above all other considerations. Such a misinterpretation of minimalist principles has led to documentation with incomplete steps and insufficient or nonexistent overviews, which are needed to provide a context and outcome for action. Even when the minimalist principles are applied correctly to documentation, the minimalist model must be expanded to consider the different modes of learning (presented earlier in this chapter) and the different levels of understanding of the subject

that users have when they learn a new product. Redish lists three considerations that have implications for documentation ("Minimalism in Technical Communication" 221):

1. Users come to documentation in different modes at different times (learning mode vs. doing mode).
2. Users differ in personality and learning style (risk takers, non-risk takers).
3. Users work in different problem domains (with different products, in different domains in one product).

Also to be considered are the needs of expert users. Hackos examines minimalism in light of the needs of these users, who have expertise in their field and with other software applications ("Choosing a Minimalist Approach for Expert Users"). She concludes that these "double experts" "need to know where to begin, where to go next, what the possibilities are, and how to get out of trouble. They do not need detailed task-oriented instructions to manipulate the interface objectives, nor do they need to consult instructional information to understand the primary purposes served by the software application" (152).

Because experts use online help frequently, if (and it's a big if) they can find what they want quickly when they consult help, the online help should be designed along minimalist lines to provide just enough information for experts to get started right away without training or instruction. As well, the words on the interface must match the expert users' vocabulary and the tasks must match the users' goals. For those who aren't experts, functions such as balloon help (Macintosh System 7.0 and above) can be turned on to explain the icons and features on a toolbar to those who need such explanations.

The influence of minimalism has been profound in its expansion of our understanding of adult learners in action and in its emphasis on understanding the tasks users want to perform as the basis for writing documentation and help to support their goals. Because the guiding principle of minimalism is task orientation, minimalism supports user-centered design.

Are Web users different? Do the principles discussed in this chapter apply to them as well? In the next section, we look at the special characteristics of Web users. For more information about usability and the Web, see Chapter 9.

Understanding Web Users

In many ways, Web users are the same as users of software or hardware. They have a goal and they are task-driven to accomplish it quickly. Although all the principles of usability apply equally to the usability of a Web site, one issue—learnability, or ease of learning—is more critical for Web users than the other issues, since Web users typically spend very little time on any individual Web site; so they need to be able to "learn" the site right away. What's more, the ever-increasing addition of new users to the Web means that there is always a large population of new learners. Approximately one in 15 visitors to a Web site has been using the Web for less than one month (Sullivan, "As Simple").

Web users have unique needs in several areas. One is speed: Web users demand fast download time. They're in a hurry to reach their goal. Users rarely *read* Web pages. Like most users of hardcopy and softcopy text, they skim, scan, and skip, but their tendency to do this increases when they use the Web. Thus, effective Web pages should use scannable text with highlighted keywords, bulleted lists, meaningful headings and subheadings, one idea per paragraph, and half the word count (or less) of conventional writing (Nielsen, "Changes") However, Web pages should be designed for readability first, not to show off complicated graphics or an array of colors. Too much of either can be "visually taxing" and "chaotic" (Sullivan, "The 'Vision Thing'"). A case can be made that bland sites are preferred when the customer is seeking information at high speed, as in banking transactions on the Web (Hurst).

Web sites also need to be predictable, as users coming to new sites bring the schema they have acquired from experience with familiar sites. Contrary to research on early users of the Web, at a time when its typical features were less well known or standardized, many users will now scroll "beneath the fold," the point below the visible portion of their screen ("For Whom the Page Scrolls" and Nielsen, "Changes"). Users scroll because they are looking for an appropriate link. How well they succeed in choosing the correct link depends on how well the site differentiates between links so that the user can predict which link to choose. Because users skim text in search of information, links are harder for users to see when they are embedded within text. When users click on a link and jump to a location, usability studies show that they will frequently go back, using the "back" button, and click on another link in pursuit of their goal. This phenomenon of *pogo sticking*, or jumping around on the site, can be a serious problem for users if their goal is to compare information. Their cognitive load may not be able to sustain the effort of remembering what they learned at the first link and then retaining that information for comparison when they locate the second link. Users become frustrated when the site forces them to jump back and forth or to write down information to compare to other information at a different link ("Pogo-sticking").

When users have a goal of obtaining information from a Web site, they will ignore banner ads (rendering them ineffective) if they interfere with the accomplishment of the goal. When the banner ad contains animation, users merely block out the ad when searching for content. Gratuitous animation, on the other hand, is a real distraction to users because it makes it harder for them to read or skim the site's content. Usability Web studies found that users put their hands over the animation as they tried to work to accomplish their goal (Spool et al.). However, according to these usability studies, users have "a seducible moment" that occurs after they have accomplished their goal. This is the point at which users can be lured from the path of their original goal to the site's goal of selling something. Users seem willing to be seduced away once their goal is met. Recognizing this, some successful e-commerce sites now place advertising below information items or embedded in information blocks at a point beyond which they determine users have satisfied enough of their goal to be lured away ("Creating Seducible Moments").

When users go to a Web site to shop, ease of use is the most critical factor in their ability to succeed in making a purchase and in their willingness and desire to return to the site. The Danish E-Commerce Association conducted a survey of 2,929

Internet users in Denmark, of which 61% had made purchases on the Web. Respondents were asked to list their top five reasons for shopping on the Web. The results are shown in Table 3.7 (reported in Nielsen, "Why People Shop on the Web").

Far and away, ease of purchase was the number one consideration. However, most Internet users do not visit a site to make a purchase. They go to the site for information. As the same Danish study shows, only 5% go to a site with the intent of buying something. Once there, if they feel confident about the ease of use of the site and usefulness of the information, they may make a purchase or they may return later to make a purchase. To gain the loyalty of Web users, Web sites should be designed so that visitors will return. For Web site designers, that means "provid[ing] useful content in a format that works the way people think" (Spool et al. xiii). *The Dotcom Survival Guide* (Creative Good) characterizes the customer as driven by a particular goal when visiting a Web site. Customers do not want to experience everything; rather, they want to experience the one thing they are seeking. Driven by this goal, their behavior fits the *page paradigm*, which means that users ask the following question about the page they choose (35):

Does the page take me closer to my goal?

- YES: Click to go closer to the goal.
- NO: Click Back to try again on the previous page.

In these ways, Web users may be different from users of other interfaces and products. In the next section, we continue to examine these differences, focusing on the specific needs of older users.

Web Sites and Older Users

Older users' needs must also be taken into consideration. One study, reported in *User Interface Design Update*, found that older users (ages 64–81), as compared to younger users (ages 19–36):

- Searched less efficiently
- Had the most trouble with tasks that required three or more mouse clicks
- Had more difficulty recalling previous moves and the location of previously viewed information

TABLE 3.7 Most Important Reasons People Shop on the Web

Easy to place an order	83%
Large selection of products	63%
Cheaper prices	63%
Faster service and delivery	52%
Detailed and clear information about what is being offered	40%
No sales pressure	39%
Easy payment procedures	36%

Source: Jakob Nielsen, "Why People Shop on The Web." *Alertbox* 7 Feb. 1999 <http://www.useit.com/alertbox/990207.html>.

- Were more likely to scroll a page at a time, vs. younger users who scanned a line at a time
- Searched less efficiently, making 81% more moves than younger users

Although the study may be flawed in that the younger group had much more familiarity with the Internet than the older group, still the findings about older users' differences seem to be "memory-related" rather than the choice of navigation strategy they adopted. More work needs to be done to understand the unique requirements of an aging population. In setting up usability tests of Web sites (or any graphical user interface) that attract an older user population, it may be important to include users with bifocals (or trifocals) to get information about what special issues they may have with screen visibility and readability. While the use of bifocals or trifocals is not restricted to an older population, it becomes increasingly common as the population ages. This, too, is an issue that needs further research.

Summary

In this chapter, we have examined the nature of field studies, especially contextual inquiry, to learn about users and their tasks and goals within their own environment. We placed field studies at the starting point of user-centered design. We looked at different techniques for conducting field studies, as well as the constraints placed on conducting such site visits. We also looked at users as learners, the discipline called cognitive psychology. As part of our discussion of this subject, we focused on schema theory or users' creation of mental models to help them shape new experiences to fit a previous mental model or to create a new mental model for future use. We then looked at the unique needs of users of documentation.

Next, we addressed learning styles to understand the different approaches that people employ. We examined how users view objects, particularly with regard to the objects in a graphical user interface (GUI). We discussed four clues that objects present:

- Affordances—the actual or perceived properties of an object
- Constraints—the properties that limit an object's use
- Mappings—the properties that suggest how we should interact with the object
- Visibility—the degree to which an object conveys its affordances, constraints, and mappings

With a general overview of learning theory, we focused on the unique needs of adult learners. For important contributions about adult learners and their needs regarding documentation, we looked to the findings of minimalism, as it was originally presented in 1980, as it has come to be understood (and misunderstood), and as it can be applied today to different audiences and different interfaces. Because the Web interface has become so important and poses unique challenges for users, we looked particularly at the research on Web users to see what similarities and differences Web users have, as compared to users of other media.

What the research shows, and what we have summarized in this chapter, is that users have a critical need to feel in control, they need a good mental model that makes sense to them, and they need to maintain a sense of certainty about what is happening to feel confident about the tool they are using to accomplish a task in keeping with a goal. Yet, users frequently feel out of control and frustrated, finding the experience more difficult than it ought to be.

Coming Up

The preparation for user-centered design begins with the principles established in this chapter. In the next chapter, we look at a methodology to increase the level of understanding we have of the user's world so that we design products that are user-centered rather than feature-centered. This methodology is called *iterative* testing, which allows a product to be tested and tested again, each time building into the product what we have learned about users. Through iterative testing, we maintain a continuing dialogue with the user to reaffirm that the design matches the user's world.

Questions/Topics for Discussion

1. Describe what *mental models* are and how they should factor into designing products for users.
2. What is the meaning of field studies and why are these studies the foundation for user-centered design?
3. What is the difference between users' tasks and users' goals? Which is the more critical aspect to focus on in developing user-centered products?
4. Describe some of the techniques that can be used to gather information about users and tasks during site visits. Which do you think would be easiest to do? Which would be hardest? Explain the reasons for your choices. Which technique would be most effective? Where does it fall within your easiest-to-hardest continuum?
5. List some of the factors that make site visits difficult to do. What impact would these problems then have on the use of site visits for data-gathering about users. Can you suggest any ways to overcome these problems to increase the frequency of use of site visits as part of a user-centered design process?
6. Cooper describes the personas that can be created after a site visit. What are these personas and what are their characteristics?
7. What is a *schema* and how does schema theory affect the design of usable products? Think of an example of a schema that is incorporated into the design of a product you use. How does it help you use the product?
8. How is the given-new principle violated in the following example? "Click on the red flower pot to take you to the landscape drawing page." As you think of the given-new concept, what are users likely to do with this step in the instructions? How can you rearrange the information to put it into given-new order?
9. Using the table of learning styles presented on page 108, describe what learning style is yours. Based on your learning style, what approach do you take to learning how to use a new product?
10. How does the minimalist approach to documentation support adult learning theory?
11. In what ways are Web users different from users of products in other media? In what ways are they similar? How does information presented on the Web need to be designed to account for the differences?

Exercises

1. Requiring users to use the "trash can" to delete a diskette in the Macintosh interface and the "Start" option to locate "Shut down" in the Windows 95 interface violated mental models that users have. Identify another example of a violation of a mental model and describe the way in which it causes problems.
2. Vocabulary needs to be clear and match the user's vocabulary. What is wrong with the following three choices available to users in a computer program:

   Log in
   Open
   Create

   Describe how you think each command differs from the others. Now, redesign the commands to make the choices clearer to users.
3. Conduct a task analysis of a common task you perform but that others may not know how to perform. Examples include such common tasks as setting the alarm on your watch, picking up a message on your pager, or forwarding a phone call in a voice-mail system. Do the following to perform the task analysis:
   - Write down each step in the task.
   - Write down what each user would be expected to perform for each step in the task.
   - What down a brief description of your primary user for this list of tasks.
   - Are there any differences in the tasks for a different type of user? For instance, if your primary use is a novice, would some tasks be different for expert users? Write down a list of any of the steps that would be eliminated or done differently for expert users.
4. Using an available product in the classroom, such as an overhead projector or computer monitor or VCR, focus on one usability aspect of the prod-

uct, such as the location of a button or switch or the shape of a part of the product, and create a list of issues that need to be considered as part of user and task analysis. This activity can also be done in groups, with each group taking a different issue for the same product.

For Your Project

Plan a site visit to observe users working with the type of product you will be testing. The site could be someplace on campus or in the office or at home. For instance, you could observe someone using a word-processing program or spreadsheet. Or, you could observe someone using your type of product to cook a meal or to record a program using a VCR. Describe an objective for the site visit and your plan for the visit. Include a list of questions you will want to have answered, either through direct observation or an interview with the user, as part of the site visit.

References

Barry, Dave. "In any Tricky Battle of Wits with Appliances, I'm Toast." *Atlanta Journal-Constitution* 27 Feb. 2000: D8.

Carlton, Jim. "Computers: Befuddled PC Users Flood Help Lines, and No Question Seems to Be Too Basic." *The Wall Street Journal*, 1 Mar. 1994: B1.

Carroll, John M. "Reconstructing Minimalism." *Minimalism Beyond the Nurnberg* Funnel, Ed. John M. Carroll. Cambridge, MA: MIT Press, 1998. 1–17.

Carroll, John M., and Hans van der Meij. "Ten Misconceptions About Minimalism." *Minimalism Beyond the Nurnberg Funnel*, Ed. John M. Carroll. Cambridge, MA: MIT Press, 1998. 55–90.

Coe, Mariana. *Human Factors for Technical Communicators*. New York: Wiley, 1996.

"Contextual Enquiry." Usability Techniques series. Information &Design, 25 Nov. 1999 <www.infodesign.com.au>.

Cooper, Alan. "Goal-Directed Design." *Dr. Dobb's Journal* 25 June 1996. 21 June 1999 <http://www.cooper.com/articles/drdobbs_goal_directed.html>.

———. *The Inmates Are Running the Asylum: Why High-Tech Products Drive Us Crazy and How to Restore the Sanity*. Indianapolis, IN: SAMS, 1999.

"Creating Seducible Moments." *UIEtips* 25 Feb. 1999 User Interface Engineering <http://www.uie.com>.

Creative Good. *The Dotcom Survival Guide* 12 June 2000 <www.creativegood.com>.

"Drag and Drop Has a Learning Problem." User Interface Engineering's *Eye for Design*, Jan.–Feb. 1996: 3, 5.

Ehrlich, Kate, and Janice Anne Rohn. "Cost-Justification of Usability Engineering: A Vendor's Perspective." *Cost-Justifying Usability*. Eds. Randolph G. Bias and Deborah J. Mayhew. Boston: Academic Press, 1994: 73–110.

Faulkner, Christine. *The Essence of Human-Computer Interaction*. Englewood Cliffs, NJ: Prentice Hall, 1998.

"For Whom the Page Scrolls." *UIEtips* 20 Mar. 1998 User Interface Engineering <http://www.uie.com>.

Gunn, Robi. "Field Trials: Trials and Tribulations of a Field Visit." *Usability Interface*, Oct. 1998 <http://www.stcsig.org/usability>.

———. *User and Task Analysis for Interface Design*. New York: Wiley, 1998.

Hackos, JoAnn T. "Choosing a Minimalist Approach for Expert Users." *Minimalism Beyond the Nurnberg Funnel*, Ed. John M. Carroll. Cambridge, MA: MIT Press, 1998. 149–77.

Hackos, JoAnn T., and Janice C. Redish. *User and Task Analysis for Interface Design*. New York: Wiley, 1998.

Hurst, Mark. "The Web's Identity Crisis." 21 Jan. 2000 Creative Good <http://www.goodexperience.com/columns/012100.identify.htm>

Janah, Monua. "AuctionWatch Site Redesign Sets Off a Revolt." *San Jose Mercury News* 18 Aug. 1999 <http://www.mercurycenter.com/business/top/054559.htm>.

Knowles, Malcolm. *The Adult Learner: A Neglected Species*, 3rd ed. Houston: Gulf, 1984.

Laskas, Jeanne Marie. "Not His Typing." *Washington Post* 10 Oct. 1999: W31.

"Making Online Information Usable." 20 Oct. 2000 User Interface Engineering <http://www.uie.com/online.htm>.

Miller, George A. "The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information." *Psychological Review* 63.2 (1956): 81–97.

Nielsen, Jakob. "Changes in Web Usability Since 1994." *Alertbox* 1 Dec. 1997 <http://www.useit.com/alertbox/9712a.html>.

———. "How Users Read on the Web." *Alertbox* 1 Oct. 1997 <http://www.useit.com/alertbox/9710a.html>.

———. "Novice vs. Expert Users." *Alertbox* 6 Feb. 2000 <http://www.useit.com/alertbox/20000206.html>.

———. "The Paradox of the Active User." Sidebar in "Personalization is Overrated" *Alertbox* 4 Oct. 1998 <http://www.useit.com/alertbox/activeuserparadox.html>.

———. "Tech Support Tales: Internet Hard to Use for Novice Users." *Alertbox* 1 Apr. 1997 <http://www.useit.com/alertbox9704a.html>.

———. "What Do Users Really Want?" *International Journal of Human-Computer Interaction* 1.1 (1989): 137–47.

———. "Why People Shop on the Web." *Alertbox* 7 Feb. 1999 <http://www.useit.com/alertbox/990207.html>.

Norman, Donald A. *The Design of Everyday Things*. New York: Doubleday, 1988.

———. *The Invisible Computer: Why Good Products Fail, the Personal Computer Is So Complex, and Information Appliances Are the Solution*. Cambridge, MA: MIT Press, 1998.

"Pogo-sticking: Downs and Ups." *UIEtips* 13 July 1998 User Interface Engineering <http://www.uie.com>.

Redish, Janice. Checklist for Planning Materials for a Site Visit <http://www.stcsig. org/usability/resources>.

———. "Minimalism in Technical Communication: Some Issues to Consider." *Minimalism Beyond the Nurnberg Funnel*. Ed. John M. Carroll. Cambridge, MA: MIT Press, 1998. 219–45.

——— "Understanding Readers." *Techniques for Technical Communicators*. Eds.