

Reading to Learn to Do*

JANICE C. REDISH

Abstract—Classroom reading primarily involves reading to learn or reading that centers on topics, whereas workplace reading primarily involves reading to do or reading that specifies actions. Redish discusses her development of “reading to learn to do” materials or tutorials that help users both to use a program and to learn. Her comparison to the research of Carroll and colleagues and her guidelines for developing effective tutorials provide help for the technical communicator’s design of tutorials and implications for the educator’s design of assignments.

IN THE LATE 1970s, Tom Sticht and his colleagues, studying literacy problems in the Army and Navy, found it useful to distinguish two types of reading tasks, which they called reading to do and reading to learn. [1, 2] When reading to do, the reader’s primary goal is to extract information for immediate action. As Sticht says, “once located and applied, the information can be forgotten. Because the information is stored in the book, it does not have to be ‘stored in the head’; it can simply be looked up if needed again.” [3] Reading to do materials include instructions for installing, operating, and maintaining equipment; decision memos; employee benefits handbooks and insurance policies; and forms to be filled out.

By contrast, when reading to learn, the reader’s primary goal is to absorb information for future recall. As Sticht says, in a reading to learn task, readers employ various study strategies to extract and retain information “in the head.” Reading to learn materials include textbooks, journal articles, and essays. [4]

Sticht and his colleagues found that students in school spend most of their time reading to learn while workers on the job spend most of their time reading to do. Other research in industrial settings has confirmed both the value of distinguishing between reading to learn and reading to do and the fact that most workers, most of the time, read to do. Mikulecky’s research in schools and in industrial settings [5] strongly supports Sticht. Sticht’s and Mikulecky’s conclusions are that only 15 percent of what students read in school is reading to do. On the job, the situation is reversed: only 15 percent is reading to learn.

The projects that my colleagues and I have been doing at the Document Design Center for the last 10 years further confirm Sticht’s and Mikulecky’s findings. Helping writers in a variety of businesses and government agencies, we have

Janice C. Redish earned her doctorate at Harvard University and is currently a Vice President of the American Institutes for Research and Director of AIR’s Document Design Center.

*Most of this paper previously appeared in *The Technical Writing Teacher* XV, 3 (Fall 1988), pp. 223–233, and is reproduced with the permission of the Association of Teachers of Technical Writing.

seen that people at work read to do far more often than they read to learn. Moreover, people in their daily lives read to do far more often than they read to learn. As we discovered in a series of studies in the late 1970s and early 1980s, the documents that are critical to people in the course of their lives are action documents—from income tax forms to college loan applications to the instructions for putting together a VCR. [6]

These findings about two types of reading would be interesting but not terribly important if it weren’t for two other points:

- Students aren’t learning how to work with reading to do materials.
- Students aren’t learning how to write so that others can read to do.

Many writers in the workplace do not appreciate the distinction. They are used to seeing reading to learn materials, and, therefore, they write as if their readers are reading to learn when they should be writing for readers who are reading to do. They write descriptions when people need instructions for immediate action.

Compare, for example, the two sets of passages in figure 1.

The major difference between the before and after in each set of passages is that the first talks about the topic, the second specifies actions. The first is based on a reading to learn model, the second on a reading to do model. Most of our work is helping companies turn material that was written like a textbook into material that works for busy people who are trying to get a job done.

DEVELOPING READING TO DO MATERIALS

Reading to do materials are relatively easy to develop when the problem domain is simple. For example, if the problem is to set up your new computer, the best documentation is a step-by-step manual that gives a series of pictures and instructions, as in figure 2. [7]

Although more research is needed, extensive work has been done on how to construct an effective job performance aid. (See reference 8 for a review.) My group and others have shown how to design and write effective workplace documents like insurance policies, regulations, employee benefits handbooks, and instructional manuals. [9–12]

But what should we do when the problem domain is not clear

Before:	The END command is used to terminate the program.
After:	When you want to leave the program: 1. Type END 2. Press ENTER
Before:	Installation of the program on the hard disk is a necessary procedure that must be accomplished before the program can be used.
After:	You must first install the program on your hard disk.

Figure 1. The Difference Between Description and Instruction in a Computer Manual

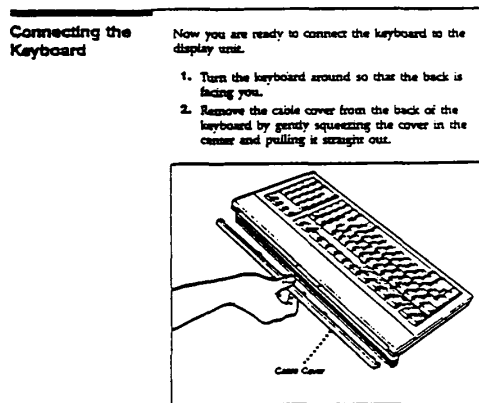


Figure 2. Page From a Useful Reading-to-Do Manual

and straightforward? What should we do when the document does not fit neatly into one of the two basic categories?

DEVELOPING "READING TO LEARN TO DO" MATERIALS

Not all documents fit neatly into Sticht's dichotomy. Consider, for example, the case of a tutorial for learning how to use a computer program. Researchers and technical writers who deal with computer documentation have been concerned with how best to construct tutorials to maximize their use and effectiveness. I suggest that we may be able to understand the problem domain better if we think of tutorials as a third type of reading task, one that we might call "reading to learn to do."

A tutorial is different from the user's guide or the online help screens for a computer program. An effective user's guide, or online help system, should be an example of reading to do. When users turn to the user's guide or the help system, they are usually trying to solve a particular problem or to get a specific job done. The key to a successful user's guide or help system is to let the user get in, get the information, and get out as quickly as possible.

The user who sits down with a computer tutorial is in a different mode from the user who goes to a user's guide while in the middle of a task. The user comes to the tutorial both to do (that is, to use the program) and to learn. A tutorial user is

almost certainly new to that particular program, whether the user is a novice or an expert at the computer.

A tutorial should help users to

- Gain a basic understanding of the concepts and structures of the program
- Become comfortable with the program so that they will want to continue to use it and be satisfied with it
- Be able to perform basic, relevant tasks with the program after using the tutorial
- Transfer what they have learned from the examples in the tutorial to other situations that were not directly covered in the tutorial

Users should do the tutorial and also learn from it, not to recite procedures or describe the system afterwards, but to do their own tasks effectively and efficiently with little or no later help.

How should we construct a tutorial so that users can learn to do? Relevant research in this area has been conducted by John Carroll and associates at IBM's Watson Research Center [13-18] and by Davida Charney and Lynne Reder at Carnegie-Mellon University. [19-21] I'd like to discuss some of their findings and add to them from our work at the American Institutes for Research (AIR), where we develop tutorials in our Document Design Center and test tutorials in our Usability Test Laboratory.

The research has shown that treating "reading to learn to do" materials like traditional reading to learn textbooks doesn't work. Tutorial users will not read long prose passages, advanced organizers, or prose summaries. Although we want them to understand the concepts and structures of the program, we have to build that knowledge through their use of the product, not by giving them pages and pages to read. Furthermore, tutorial users want to accomplish realistic workplace tasks. Giving users "drill and practice" on basic system-driven skills without the context of a realistic task often leads to unhappy and unsuccessful users. [18] That is, users want to learn how to *edit a paragraph* not how to *move the cursor to the top of the screen*. They can learn about moving the cursor while learning how to edit a paragraph.

The research has also shown that treating "reading to learn to do" materials like traditional reading to do job aids doesn't work either. When tutorial users just follow the procedural steps without understanding why they are doing them, they may succeed in completing the tutorial and then be unable to recall the procedures later or transfer them to other situations. In fact, in a recent study of tutorials for a spreadsheet program, Charney and Reder found that learners who just typed what they were told learned no more than learners who just read through the tutorial and examples. [21]

On the one hand, tutorial users are similar in many ways to readers of reading to learn materials. The user is an ac-

tive learner who brings his or her own goals, strategies, and schemas to the tutorial. That is, the top-down view of reading and readers is as applicable to computer tutorials as it is to textbooks. Carroll and Mack found that tutorial users create hypotheses about what is happening as they go along, and they act on the basis of their hypotheses, sometimes refusing to accept the text or screen when it contradicts their current hypothesis. [14]

On the other hand, the user of a computer tutorial is also similar in many ways to the user of reading to do materials. Users are invariably busy people. The computer is a tool for them. Their job is to produce work, not to read computer manuals. Therefore, when doing a computer tutorial, users want to do relevant work right away. They will bypass paragraphs of text. Carroll also found that his subjects acted as soon as they saw anything about an action—even if it was in an introductory prose section. [13]

In a series of studies of novice computer users learning a word processing program, Carroll's group at IBM [13] found that learners:

- Are easily overwhelmed
- “Jump the gun” to act
- Skip around to work on what is important to them
- Create hypotheses and act on them rather than reading and following directions
- Sometimes ignore the screen when following a tutorial step-by-step
- Make errors and have trouble recovering from them
- Want to do real work and want to understand *why* as they proceed

In working with users as we develop and test tutorials at AIR, we have found many of the same characteristics of learners as Carroll did. From our work, the most critical factors seem to be that learners

- Do not want to read
- Want to accomplish something relevant quickly
- Want to be actively involved in the learning
- Want to understand *why* and *what* they are doing

We have also found that there are two types of learners: those who want to play with the program and who are happy to try out their own examples, and those who want to have their hands held and who are only comfortable when we give them specific examples to use. Furthermore, we have found that we cannot assume that learners know what they want to do. Many computer products offer possibilities beyond the limits of the problem domain as the user defines it before using the product. That is, with a computer product, we are often trying to give users new ways to do old tasks (typing a letter, for example) and we are also introducing them to new tasks—options that they didn't even know they had (changing the type size in a heading, for example). An important aspect of tutorials may be to expand the user's task domain.

Carroll's recommendations for meeting the needs of computer learners are to give the users “training wheels” and to create “minimal manuals.” By training wheels, Carroll means a real but simpler system for the user to learn on. You can do this by having only parts of the product available to the user for the tutorial. In that way, new users can work with the relevant parts of the actual product without getting hopelessly lost when they make a mistake. By minimal manual, Carroll means “slashing the verbiage,” eliminating all repetition, summaries, reviews, practice exercises, indexes, and appendixes. He also means making sure that the approach is task-oriented (the hallmark of good reading to do materials) and that the tutorial includes error recovery procedures.

In his early studies, Carroll advocated a guided exploration approach. [15] For the guided exploration, Carroll and his associates developed a set of cards that were truly minimal. Each presented a relevant user's task, gave the user a few hints about starting the task, included some checkpoints so the user would know if he or she was on track, and included information about error recovery. In a comparison test with the original, commercial manual, users who learned with Carroll's cards spent half as much time in learning, committed half as many errors, recovered more often from their errors, and did much better in transferring the learning to other tasks. [18] Nevertheless, some of the users felt uncomfortable with the guided exploration cards and missed having a real manual.

Because the differences between the original, commercial manual that Carroll was working with and the guided exploration cards or the minimal manual are so great, it is not clear which features of the minimal manual are ideal. A truly minimalist approach—with no previews or reviews, with no explanations or elaborations—may not be best in all situations. For example, Reder, Charney, and Morgan found that for users who are new to a system and do not have clear goals of their own, some elaboration of the text is more helpful than text with no elaboration. [19] In a recent study, Charney and Reder found that learners who were given a problem, told to try to solve it, and then shown a solution, learned best. [21]

The approach that we have taken at AIR in developing tutorials since 1984 slashes the verbiage as Carroll's does, but also includes some instructional design elements that provide minimal previews and elaborations. To meet both *learning* needs and *doing* needs, we have found that the following guidelines work well in developing effective tutorials:

- We break the entire tutorial into short lessons (three to five pages) and allow the learner to stop working at the end of any lesson and to reenter the tutorial at any lesson (figure 3).
- We begin each lesson with a signpost section, a bulleted list of the tasks that the lesson covers (figure 4). This list provides a preview, but it can be scanned quickly—no sustained reading is required. Because it is a task list (not a description of the steps), users do not jump the gun to do the actions at this point. We find that this short list of tasks

Chapter 9 in *Using Personal Card File* tells you more about finding cards and suggests many other ways of using this feature.

When you are ready to continue, go to the next session, which lets you use cards to make a mailing list.


(If you prefer to stop, just touch )

Figure 3. Let Users Choose Whether to Continue or Stop at the End of Each Lesson.

4. Selecting, Changing, and Printing a Card

With PCF you can keep your cards up to date without messy erasing or crossing out. And, if your computer has a printer attached, you can make printed copies of your cards. In this session you will:

- Select a card
- Move the cursor to the information you want to change
- Change the information on a card
- Print a copy of a card

-
-
-

Selecting a Card Let's say that Mr. de Rocha of the Brazilian Coffee Corporation has a new telephone number, and you want to put the new number on the card.

Figure 4. Signpost Preview and Subheading in a Tutorial

provides a framework for users to feel comfortable with the tutorial.

- We then use each item in the signpost list as a subheading at the appropriate place in the lesson. Again, this provides an advanced organizer that requires no sustained reading. Because these task-oriented headings also become the table of contents for the tutorial, learners can quickly find relevant sections if they want to skip around in the tutorial.
- We are careful to provide a very brief motivation for an action before suggesting the action (figure 5).
- We try to use metaphors and examples that are meaningful to the learners. Without the brief motivating statement that puts the recommended action into an appropriate context, users are just typing or pressing keys and not being actively involved. In their recent work, Carroll and his associates found that it is sometimes important to explain why users must do something and that analogies to the users' outside experiences are helpful.
- We find that inviting learners to use their own examples if they want, but also providing one for learners who are not comfortable making up their own, meets the needs of different individuals (figure 6). In the same vein, inviting users to practice lets users decide whether to skip the practice or to spend time playing on their own.
- We try to build from the simple to the complex. This means that the organization of a tutorial may well differ from the organization of the companion user's guide. For example, in the user's guide for a complex data base system, we cover *creating a file* before *editing a file*, but in the tutorial we

You can also search for cards that have the same number in a given field. For example, you can search for all the cards that have the same date or the same zip code.

Let's suppose you want to know which companies you last heard from sometime in 1983.

Figure 5. Giving Users a Reason Why They Might Want to Learn a Particular Feature

You can experiment with finding cards by searching for words in other fields. For example, if you have forgotten Mr. Menard's first name, you can type *menard in the Agent's Name field and find his card.

Figure 6. Giving Users Freedom to Experiment While Also Giving Them an Example to Follow if They Want One

supply a file for learners to edit before we have them create a new file.

- We always have the learners accomplish something in each lesson. We may want users to learn concepts and techniques; they want to do real work.

In 1984, we used these guidelines (and others about good layout and clear writing) to develop a new tutorial for a small data base product to be used on a personal computer. The original tutorial consisted of a few, very long lessons with no previews or reviews, with no motivation for the steps that learners were told to take, and with a running example that was not meaningful to many users. In a comparison usability test of the original and new tutorials, learners completed the new tutorials more successfully and were happier with them. And more importantly, the learners who had the new tutorials were far more successful at transferring the learning to other tasks afterwards.

In other cases, our successful tutorials have not been so minimal. In very complex products where the user's guide is several hundred pages long, users may continue to refer to the tutorial for some time after they have worked with it. In that case, we have found that the tutorial may continue to help users if it keeps some of the features that Carroll suggested be dropped from the minimalist manual.

For example, in writing tutorials for a very large, command-driven, mainframe, data base product, we ended each brief chapter with a summary. The summary was a table of the commands that the learner may have used in that lesson and what the commands do. The table summarizes the lesson without giving the user prose paragraphs to read. It looks like a quick reference card. For this very complex and function-rich product, many new users go back to the tutorial summaries for some time after they begin using the product, before they are willing to pick up the user's guide. For the same product, the learners wanted an index in the tutorial manual. Even though the headings in the tutorials are all user-relevant tasks (writing a memo, selecting the data for a pie chart), some users

prefer an index as their primary access to a manual—even to a tutorial.

Carroll's Minimalist Training Model [18] is one of iterative development. The tutorial writers develop a minimal training manual, try it out with users, and revise it in as many cycles as is necessary. As Carroll says, only with iterative development and input from users can the writers know which errors users make and, therefore, what error recovery procedures to include. Only with iterative development can the writers know how much elaboration is needed for each task and what metaphors are meaningful to the audience. The iterative development, however, should not stop when the product is delivered. Almost all computer products go through many versions (releases), and we have found that the ways that users use tutorials over time in real settings sometimes differ from the expectations that we would have from watching users in the laboratory.

CONCLUSION

Sticht's division of documents into reading to do and reading to learn has been very useful to researchers, educators, and writers. However, some documents don't fit neatly into one or the other category, but are best seen as hybrids of the two. In particular, computer tutorials constitute a type of "reading to learn to do" that has elements of both reading to learn and reading to do.

Thinking about computer tutorials as "reading to learn to do" can help us appreciate the findings of recent research on tutorials. In particular, tutorials work best when users are invited to act rather than to read, when users are involved as active learners, when the tutorial focuses on tasks that are relevant to users.

The approaches described in the work of Carroll and his colleagues, of Charney and Reder, and of the Document Design Center are similar, but not exactly the same. Many interesting research questions remain. We do not know which features of these three approaches work best for which types of users and which types of computer programs. Tutorials are still a fascinating area for research on a document that combines aspects of reading to learn and reading to do.

NOTES AND REFERENCES

1. Sticht, T., *et al.*, *The Role of Reading in the Navy*, San Diego, CA: Naval Personnel Research and Development Center, 1977.
2. Sticht, T. and Welty-Zapf, D., *Reading and Readability Research in the Armed Services*, Alexandria, VA: HumRRO, 1976.
3. Sticht, T., "Understanding Readers and Their Uses of Text," in *Designing Usable Texts*, T. M. Duffy and R. Waller (ed.), Orlando, FL: Academic Press, 1985.
4. Note that Sticht and his colleagues were studying only nonfiction. Novels, stories, poetry, and plays are outside the domain that I am discussing in this paper.
5. Mikulecky, L., *Job Literacy: The Relationship Between School Preparation and Workplace Actuality*, Bloomington, IN: Indiana University, 1981.
6. Rose, A. M., "Problems in Public Documents," *Information Design Journal* 2 (1981), pp. 179-96.
7. This and all later figures are from manuals for Hewlett-Packard products that the Document Design Center at AIR helped to develop. The excerpts are used with permission of Hewlett-Packard.
8. Smillie, R. J., "Design Strategies for Job Performance Aids," in *Designing Usable Texts*, T. M. Duffy and R. Waller (ed), Orlando, FL: Academic Press, 1985, pp. 213-243.
9. Redish, J. C., "Writing Regulations (and Other Legal Documents) in Plain English," *Legal Notes and Viewpoints* 1 (1981), pp. 73-93.
10. Redish, J. C., *Beyond Readability: How to Write and Design Understandable Life Insurance Policies*, Washington, DC: American Council of Life Insurance, 1984.
11. Redish, J. C., "Writing and Designing Effective Software Manuals," paper based on a workshop for the National Computer Graphics Association, June 1986.
12. Redish, J. C., Battison, R., and Gold, E., "Making Information Accessible to Readers," in *Writing in Nonacademic Settings*, L. Odell and D. Goswami (ed.), New York: Guilford Press, 1985, pp. 129-153.
13. Carroll, J. M., "Minimalist Training," *Datamation* 30 (1984), pp. 125-36.
14. Carroll, J. M. and Mack, R. L., "Learning to Use a Word Processor by Doing, by Thinking, and by Knowing," in *Human Factors in Computer Systems*, J. C. Thomas and M. L. Schneider (ed.), Norwood, NJ: Ablex, 1984, pp. 13-51.
15. Carroll, J. M., *et al.*, "Learning to Use a Word Processor by Guided Exploration," research report from IBM Watson Research Center, 1984.
16. Carroll, J. M. and Carrithers, C., "Blocking Learner Error States in a Training Wheels System," *Human Factors* 26 (1984), pp. 377-89.
17. Catrambone, R. and Carroll, J. M., "Learning a Word Processing System With Training Wheels and Guided Exploration," *SIG + GI, Proceedings of the ACM Conference on Human Factors in Computing Systems*, 1987, pp. 169-74.
18. Carroll, J. M., Smith-Kerker, P. L., Ford, J. R., and Mazur-Rimet, S. A., "The Minimal Manual," *Human Computer Interaction* 3 (1987-1988), pp. 123-55.
19. Reder, L., Charney, D., and Morgan, K., "The Role of Elaborations in Learning a Skill From an Instructional Text," *Memory and Cognition* 14 (1986), 64-78.
20. Charney, D. and Reder, L., "Designing Interactive Tutorials for Computer Users," *Human Computer Interaction* 2, 297-317.
21. Charney, D. and Reder, L., "Research on Manual Design," *Technical Communication* 34 (1987), pp. 52-53.

To the man with an ear for verbal delicacies—the man who searches painfully for the perfect word, and puts the way of saying a thing above the thing said—there is in writing the constant joy of sudden discovery, of happy accident.

Henry L. Mencken