DESIGNING EFFECTIVE DOCUMENTATION

William W. McMillan
Gayle M. Lambert
Andrew R. Jennings Computing Center
Case Western Reserve University
10900 Euclid Avenue
Cleveland, Ohio 44106
(216) 368-2982

Abstract

We borrow from research in educational psychology and document design, and from intuitive guidelines to make suggestions about the design of computer documentation. We discuss general characteristics of design, specifics, and different techniques for empirical evaluation of documents.

Introduction

Many people recognize the need for better and more systematic documentation in computer science (e.g., Peterson, 1981; Snyders, 1981), including user services professionals who must create, evaluate, and recommend user documentation. This paper has two purposes: 1) to encourage creators of documentation to use the best available technology in designing user manuals, and 2) to point out the value of, and need for, empirical research on design of principles for documentation.

There are two major sources of research on principles relevant to the design of computer documentation. One is the programmed instruction (PI) movement within education. Those who advocate PI argue that the way to teach something is to analyze it, create instructional objectives, ensure that the learner has necessary prerequisites, tailor instruction to the individual learner, assess learning, and recycle if necessary. We attempt here to apply principles of PI (and other work in educational psychology) to the design of ("non-programmed") computer documentation.

The other major source of research on design principles is the relatively young field of document design. The Document Design Project (funded by the National Institute of Education) carries out research on design principles for government and legal documents. The research is often based on, or related to, research in cognition, psycholinguistics, and PI. We offer some document design principles here in the hope that they prove useful in creating computer documentation.

The suggestions we make for designing computer documentation are derived from the research sources described above, from texts on technical writing, and from our own experience creating documentation at an academic computing center. Mixing research results with intuition is a dangerous practice; the scientific aura of the research tends to lend the intuitive principles undue respectability (Sheil, 1981). Consequently, we label claims derived from empirical research by citing references in our discussion. Principles without citations are from our experience, from the

see page ii

writing texts listed in our bibliography, and/or from intuitive arguments for PI.

When you create documentation, you make many design decisions, explicitly or by default. Below is a list of the questions that you must answer and suggestions for how to answer them. First we handle questions of general design, then specifics of design, and finally review and evaluation.

General Design

• What are your resources?

Consider what kind of documentation you can afford to create. Do you have the staff (measured in number, skill, and willingness) to create the kind of documentation you would like? Do you have the money to produce the graphics, type fonts, bindings, etc. that you would like?

If yours is like most computing centers, you do not have enough resources of any kind to produce genuine PI similar to Smock's (1980). Nor do you have the kind of control over type fonts and graphics that a professional printer has. You probably have the equivalent of one half-time writer, a reasonable text processing system, simple graphics, and Xerox facilities.

Our suggestions are to build as many PI concepts into your writing as possible, construct every fragment of prose according to good design principles, and do not worry very much about fancy graphics and layouts.

• Who writes?

We all know technically sophisticated people who could not write effective instructions for opening a can of beans to save their lives. Given user documentation created by a person with such a distribution of talents, you can red ink your way to better sentence construction and word choice, but you will probably still have an ineffective document. Assumptions about readers' abilities may change from section to section, the document may not fit into a larger documentation system, and it may have other gross defects. Red ink will not save you from violations of most of the guidelines given here. (Unless you rewrite the whole thing in red ink -- which means you should have written it to begin with, probably in another color.)

The person to write a document should be someone whose heart, soul, and talents can effectively be invested (even if only part-time) in accomplishing the kinds of things we are suggesting here. And, as many people have pointed out, the writer should be involved in documenting the system from its early stages of development. (In fact, if the writer has a good understanding of how to communicate with the target audience, he or she should probably be consulted about the design of the system itself.)

• How do you plan?

Spend a good deal of time planning and organizing. Ninety-five percent of a sample of professional technical writers queried by McKee (1972) said they use some form of outline to guide their writing. It is unclear whether planning and outlining *cause* more effective writing, but they seem to accompany it.

Brainstorming, jotting down random ideas, may be very helpful before outlining. If you set your ideas too soon in the Roman numeral concrete of an outline, you may exclude many good ideas. Atlas (1979) showed that novice writers follow an outline too rigidly and fail to revise it when necessary.

Planning should include considerations of objectives, audience, prerequisites, and other features besides content. Following sections amplify on this.

• What are the objectives?

Choose specific objectives. A fundamental step in designing PI is to specify objectives of instruction. What, exactly, is the learner to come away with? Many people in the field recommend "behavioral" objectives, i.e, a list of overt, well defined actions that the learner will be able to perform after instruction.

Is the reader to learn how to link an external subroutine library to his or her program or just that such things are possible? Is the reader to learn enough about what an operating system is to give a simple definition or just enough to identify commands that are "operating system commands?"

Telling the reader the objectives at the beginning of a section can also help. Specific objectives are better that general ones (Kaplan, 1976). It is most important to tell the reader the objectives when the organization of the document's contents is not readily apparent to the reader (Duchastel, 1979).

• What prerequisites are assumed?

Decide what abilities and knowledge the reader should have to enjoy success with the document and write it so that readers without those prerequisites will be diverted elsewhere. In bona fide PI, a pretest may be given to the learner and, if he or she does not have necessary prerequisites, the program will branch to appropriate exercises or lessons. Most of us are not in the position to create real PI, but we can write our prose to accomplish essentially the same things.

List the prerequisite abilities, such as being able to sign on to the system, compile a COBOL program, edit a file, or invoke SCSS, let the reader decide whether or not he or she has the prerequisites, and tell where to acquire those abilities if they are lacking. It is not enough to write, "We assume you have read manual XYZ." Who cares whether the learner has read that manual or not, as long as he or she has the prerequisites? You can read the manual and not get them or get them without reading the manual.

It can be a surprising exercise to list the concepts necessary to understand a document. Even for introductory documents, we may assume that the reader knows definitions of computer, file, program, operating system, terminal, default, argument, parameter, word, ASCII, timesharing, batch, command level, crash, disk, package, and so on. These are often bad assumptions. Certainly, it is not necessary to enumerate and teach all possible computer-related concepts, but we chould give attention to more than we typically do. (Burying definitions in a glossary may or may not be a solution, depending on how the reader uses the glossary, if at all.)

• What is the audience?

Define your audience and consider its characteristics. Consideration of the audience is one of the most important practices of an experienced writer. One reason that novices fail to give proper attention to their audience is that they follow outlines too rigidly (Atlas, 1979).

The audience for computer documentation, especially at an academic computing center, is probably one of the most difficult to write for. A manual may be read by people who have never used a computer and by new speakers of English, by people only slightly involved in computing and by people whose futures hinge on their success in computing, by people who will use the system to write poetry and by people who will solve non-linear programming problems. The writer must tread a fine line between insulting the reader and scaring him or her away.

Two practices improve writing for a varied audience: 1) Clearly label sections that are appropriate for only a subset of your audience so that the rest of the audience can skip them, e.g., "What is a program?" or "Calling assembly language routines." (Your introduction can also steer different readers to different sections.) 2) If explanations for a subset of your audience are not labeled, make them as succinct as possible. A systems programmer will probably not be put off by a one sentence definition of a file in an introductory paragraph, nor a political science student by a few words about memory registers.

• How will the document be used?

As you create a document, continuously imagine the reader using it. Is he or she sitting at a terminal debugging a program; sitting by the fireplace on a winter's evening, content with your prose for hours on end; referring to small sections while writing code; amidst disembodied hardware pursuing a glitch?

Manuals fall roughly into two categories: fireside manuals such as the SCSS manual (Nie et al., 1980), which is beautifully written for the patient, motivated reader; and ready-reference manuals, written for the reader who never reads introductions and loves to copy examples.

Many manuals should be usable either way. That means that explanations in the text must often be somewhat redundant with notes in examples and with command summaries. If a manual is only for one type of use, especially if it is only for reference, let the reader know in the title or introduction.

• What relation does the document have to other documents?

Design and communicate meaningful relations between documents. Plan documents so that the prerequisites for each one (except an introductory manual can be learned from another. Anticipate common reading orders, e.g., first the intro manual, then the editing manual, then the text processing manual. A system can be pieced together out of vendor manuals, textbooks, your own manuals, and on-line help files.

Give the reader a road map of the documentation system by referring to prerequisite and more advanced manuals in the documents you create. An intelligent information retrieval system would be invaluable.

• What is the sequence of the information presented?

Break material into chunks that the reader can manage easily and sequence them so that explanations build on previous explanations. For some kinds of prose, segments of about 20 lines are optimal (Frase, 1967). Perhaps logical segmenting and sequencing seem so obvious that we forget to give them very much attention. Instead, we tend to write things down in the order in which they occur to us, i.e., perform a personal mind dump. Champions of PI are careful to ensure that success with each section is likely and that prerequisites are taught before they are needed. There is no reason why a writer of prose cannot do likewise. (Reference-only documents, of course, are not used sequentially and should be organized for easy reference.)

• How do you establish the context?

Use headings, explain concepts, and explain functions to provide the reader with "advance organization." People learn things best if they can relate new knowledge to old (Ausubel, 1960). Such orientation is advance organization.

With short, well-written prose passages, headings have not been shown to aid reader comprehension (Charrow and Redish, 1980). In lengthy, technical prose, though, it seems that they would provide advance organization. A heading should be meaningful to the person not yet familiar with the specifics of the section. For example, a heading like, "Getting a list of users on the system," is better than, "The systat command." With the first heading the reader is clued to the subject before reading, but with the second he or she probably must discover the purpose of the section while reading it.

As we discussed under "prerequisites," the reader may lack many concepts necessary to understand what you are writing about. This is naturally a good reason to explain them. But even concepts that are not strictly prerequisites may aid understanding by providing advance organization. Providing a concrete model of the computer seems to be especially helpful for new computer users (Mayer, 1981). (See our "examples" section for suggestions about teaching concepts.)

Before diving into your main explanations, tell the reader the function of the software, procedure, or hardware that is the document's main subject matter, e.g., "Runoff is a program that makes text look nice," or, "PLOT10 is a collection of FORTRAN subroutines that allows you to create graphs." Not everyone who reads a document's introduction knows why he or she should be interested in its subject matter. Briefly state functions to provide advance organization and to encourage exploration.

Specifics

• What style do you use?

Study style manuals and similar sources to develop clear writing. This is not the place for an extended discussion of writing style. See our bibliography for some good references. Below is a list that summarizes some major principles. (Many of these guidelines are supported by empirical research; see the literature review in *Simplifying Documents*, listed under "Document Design Project" in our bibliography.)

1. *Word choice*: Use the simplest words possible, do not vary the name of something just to be elegant, and eliminate unnecessary words.

2. *Sentence constuction*: Use short sentences, use simple sentence construction (e.g., subject-verb-object), avoid complex conditionals (e.g., "If you have operator privileges or you are accessing a tape drive and it is after midnight..."), use a list to present a series of items (preferably indented with each item starting on a new line), and use similar phrase structure in each list item.

3. *Tone and voice*: In instructions about how to do something, address the reader directly, using "you," "we," other personal pronouns, and the imperative voice. Use active voice instead of passive (e.g., "Your command starts the program," instead of, "The program is started by your command,") unless the subject of the sentence is relatively unimportant (e.g., "The program was converted from ALGOL.")

• Should a long document have a table of contents and an index?

Of course.

• How do you create examples?

Create examples so that the reader can learn as much as possible from them without reading anyting else. Face it: People love to learn from examples and use them for reference, no matter how beautiful your prose. Use many of them and create them to demonstrate the parts of a concept or procedure that can vary and those that stay constant. Examples should follow a clear statement of the concept or procedure. Include notes that explain what is happening in the examples.

Many concepts require only one example, but those that are not very simple often require a series of examples. A series of examples should increase in difficulty and include non-examples (incorrect commands) to clarify the boundaries of the concept (Fleming and Levie, 1978).

Example of examples of CWRUed editing commands:

	*5H	moves cursor 5 spaces to right
	*-3H	moves cursor 3 spaces to left
	*↑H	moves cursor to left end of line
incorrect:	*-↑H	not a legal command
	*\H	moves cursor to right end of line
incorrect:	*2\H	not a legal command

A special kind of example is the "scenario." In a scenario, you describe for the reader a hypothetical situation and use it to show how to carry out some procedure.

For example, "Suppose you have two files with similar names and you decide to delete one. But you mistype the file name and delete the wrong one. If you notice the error immediately, you can..." When people read, they often construct their own scenarios to aid understanding (Flower, Hayes, and Swarts, 1980).

• What can be left out?

When explaining a procedure, present initially a "minimal correct procedure," without mention of options. This helps form material into manageable chunks. Let the reader master a complete, albeit simple, procedure before getting fancy. In many cases, this is all the reader will

need to get something useful done. It at least gives him or her some footing and confidence. Then introduce variations.

It is often tempting to introduce great lists of options and variations of procedures in a document. If these variations are presented elsewhere, they may only detract from a document that has a different primary purpose. They make it more expensive to produce and buy, and (never underestimate it) more imposing.

• How do you present questions?

In tutorial documents, you may want to insert short-answer questions for the reader to answer. This increases retention of information (Rothkopf and Bisbicos, 1967). General questions should *precede* pertinent discussion and specific questions should *follow* pertinent discussion (Rickards, 1976).

Review and Evaluation

• What is the review procedure?

When other people review a document, define their responsibilities and authority clearly. Ask for the kind of review that you need: grammatical, political, technical, etc. Decide, and explain to the reviewer, whether the reviewer has the authority to reject a document with which he or she is not satisfied.

Limit review. Get review that is necessary, but do not invite superfluous cooks to help with the broth.

People rarely review a document for content, organization, level of discussion, and other general characteristics. Word choice and sentence structure are the reviewer's favorite targets. Encourage reviewers to take a step back and evaluate the overall design of the document. Their view may improve your design.

• How do you evaluate the effectiveness of the document?

Evaluate documents actively, aggressively; do not count on passive techniques such as reader comment cards. This is far easier suggested than done; you need research skills, time, other resources, and management support.

Comment cards and other passive questionnaire-type techniques just about guarantee a biased sample. The people who respond probably have some characteristics that are not shared by most of the document's readers. It is better to distribute questionnaires to a sample that you have chosen to be representative and the aggressively follow up (by phone, in person) to get responses from all or most people in the sample. This probably means a smaller, but much more meaningful sample.

Structured interviews are a good first step to evaluation. They can make you aware of issues you might never have considered otherwise. Ask how people use the document, how much they read, what was missing, etc. From there, questionnaire items and other evaluation questions will evolve naturally.

Experimentation should be used more than it is to evaluate documentation. Experiments that compare the performance of several groups of subjects whose treatments differ in one or a few key ways are fraught with

difficulties (Sheil, 1981). Especially in applied (i.e., non-theoretical) settings, experimentation might better be modeled on a technique used by researchers in human problem solving: protocol analysis (Lindsay and Norman, 1972, Ch. 14). The researcher gives a detailed account (in the form of a "problem behavior graph") of the performance of one or a few subjects. Most of us would be limited to fairly informal versions of such research, but we would gain quite a bit by observing closely the way a person reads and refers to a document, and the places where most trouble is encountered. Even if our few subjects were not representative of the reader population, we might still gain more than we would from a study that lumped together the performance of subjects in each of two groups to compare two ways of designing one characteristic of a document.

Several techniques have been designed to assess the readability of prose. Hartley, Trueman, and Burnhill (1980) tried some of these techniques, including readability formulas and subject ratings, to compare two versions of a technical passage. Because of the unreliability of some of the measures and methodological faults, it is difficult to assess the value of the techniques. More research is needed on their empirical validity.

Conclusion

Principles of PI and document design provide a rich source of guidelines for the creator of computer documentation. Even if these guidelines cannot be applied as they stand in many situations, they should serve as a source of ideas for design, make documentation professionals aware of document characteristics previously designed by default, and challenge the skeptical to answer with empirical research.

Bibliography

• Cited in text

Atlas, M. A. *Addressing an audience: A study of expert-novice differences in writing* (Report 3). Washington, D.C.: Document Design Project, December, 1979.

Ausubel, D. P. The use of advance organizers in learning and retention of meaningful verbal material. *Journal of Educational Psychology*, 1960, *51*, 267-272.

Charrow, V. R., and Redish, J. C. *A study of standardized headings for warranties* (Report 6). Washington, D.C.: Document Design Project, February, 1980.

Duchastel, P. Learning objectives and the organization of prose. *Journal of Educational Psychology*, 1979, *71*, 100-106.

Fleming, M., and Levie, W. H. *Instructional message design: Principles from the behavorial sciences.* Englewood Cliffs, New Jersey: Educational Technology Publications, 1978.

Flower, L. S., Hayes, J. R., and Swarts, H. *Revising functional documents: The scenario principle* (Report 10). Washington, D.C.: Document Design Project, March, 1980.

Frase, L. Learning from prose material: Length of passage, knowledge of results, and position of questions. *Journal of Educational Psychology*, 1967, *58*, 266-272.

Hartley, J., Trueman, M., and Burnhill, P. Some observations on producing and measuring readable writing. *Programmed Learning and Educational Technology*, 1980, *17*, 164-174.

Kaplan, R. Effects of grouping and response characteristics of instructional objectives when learning from prose. *Journal of Educational Psychology*, 1976, *68*, 424-430.

Lindsay, P. H., and Norman, D. A. *Human information processing* (2nd ed.). New York: Academic Press, 1977.

Mayer, R. E. The psychology of how novices learn computer programming. *Computing Surveys*, 1981, *13*, 121-141.

McKee, B. Do professional writers use an outline when they write? *Technical Communication*, 1972, *19*, 10-13.

Nie, N. H., et al. *SCSS: A user's guide to the SCSS conversational system.* New York: McGraw-Hill, 1980.

Peterson, P. Tech writers talk. *Datamation*, February, 1981, pp. 81-82; 84.

Rickards, J. P. Interaction of position and conceptual level of adjunct questions on immediate and delayed retention of text. *Journal of Educational Psychology*, 1976, *68*, 210-217.

Rothkopf, E. Z., and Bisbicos, E. E. Selective facilitative effects of interspersed questions on learning from written materials. *Journal of Educational Psychology*, 1967, *58*, 56-61.

Sheil, B. A. The psychological study of programming. *Computing Surveys*, 1981, *13*, 101-120.

Snyders, J. User manuals that make sense. *Computer Decisions*, April, 1981, pp. 127-128; 130; 132; 139-140; 186.

Smock, D. M. Self-instructional modules: Their development and use by user services. In *Proceedings of eighth annual ACM-SIGUCC user services conference.* New York: ACM, 1980. (ACM order no. 459800)

• The Document Design Project

To learn about the DDP or to order publications, write to

American Institutes for Research
Suite 200
1055 Thomas Jefferson St., N. W.
Washington, DC 20007

Simplifying Documents: A workshop..DDP, 1979.

A workshop text and literature review that teaches how to write clearly. Contains specific learning objectives and exercises. Oriented toward government documents, but applicable to technical writing.

Simply stated

The DDP's newsletter. A joy to read (naturally) and free.

Felker, D. B. (Ed.) *Document design: A review of the relevant research.* DDP, 1980.

Serious review of educational, cognitive, and psycholinguistic research relevant to document design.

Technical reports

Most are reports of experiments on different ways to design documents. (Write to above address for a complete list.)

• Texts on writing

These sources base recommendations mainly on the experience and intuition of the authors. Your library or book store will have many other books similar to these.

Strunk, W., Jr., and White, E. B. *The elements of style* (3rd ed.). New York: Macmillan, 1979.

A classic. Still the most direct and succinct guide to correct usage and effective style.

Baker, S. *The practical stylist* (5th ed.). New York: Harper and Row, 1981.

Oriented toward term paper writing, but has very lively and useful discussions of sentence and paragraph structure.

Crews, F. *The Random House handbook* (3rd ed.). New York, Random House, 1980.

Good general handbook, oriented toward college composition.

Houp, K. W., and Pearsall, T. E. *Reporting technical information* (4th ed.). Encino, California: Collier Macmillan, 1980.

Very readable technical writing text. Includes discussions of audience analysis, types of technical exposition, graphics, reports, proposals, and more.

Harty, K. J. (Ed.) *Strategies for business and technical writing.* New York: Harcourt, Brace, Jovanovich, 1980.

A collection of pithy and enjoyable essays on major problems in business and technical writing. Contains extensive bibliography.

• PI and educational psychology

Gagne, R. M., and Briggs, L. J. *Principles of instructional design* (2nd ed.). New York: Holt, Rinehart, Winston, 1979.

A state-of-the-art introduction to instructional design by two giants in the field. Enough references to break into the literature.

Fleming, M., and Levie, W. H. (listed under "Cited in text")

An ambitious attempt to summarize design principles that optimize the learner's perception, retention, concept attainment, and attitude change. Heavy reliance on secondary sources.

Klausmeier, H. J., and Goodwin, W. *Learning and human abilities: Educational psychology* (4th ed.). New York: Harper and Row, 1975.

One of the better general intros to educational psychology. Good summaries of principles of teaching "factual information" (Ch. 10) and concepts (Ch. 11). Extensive references.

• Research methods for evaluating documents

For the novice social scientist: Leaf through books on social and behavioral research methods. If a book is at least 1/3 statistics, set it down; it was written to teach people how to impress journal editors. The two below represent a useful type. They give practical overviews of interviewing, observing, experimenting, and other techniques. (See Ch. 14 of Lindsay and Norman, listed under "Cited in text," to learn about intensive observation of a person's "problem solving" behavior.)

Phillíber, S. G., Schwab, M. R., and Sloss, G. S. *Social research.* Itasca, Illinois: Peacock, 1980.

Sommer, R., and Sommer, B. B. *A practical guide to behavioral research.* New York: Oxford University Press, 1980.