Procedural and Declarative Information in Software Manuals: Effects of Information Use, Task Performance, and Knowledge. Nicole Ummelen. Amsterdam: Rodopi, 1997.

Reviewed by David K. Farkas
University of Washington

This book presents the elaborate and carefully reasoned series of experiments that Nicole Ummelen conducted for her doctoral dissertation at the University of Twente. It is an important contribution to our understanding of computer documentation and, by extension, instructions of all kinds. It is also admirably clear and well written.

The questions Ummelen seeks to answer are these: How useful is declarative (or, in other words, "explanatory" or "background") information in procedures? Do users choose to read this material? If so, does it help them, and how? Do the complexity of the task and the background and goals of the users affect their reading decisions, task performance, and understanding of the software?

These questions are of fundamental importance. They bear directly on the claims of some minimalist researchers that declarative information is largely superfluous. Also, these questions have direct and immediate practical importance in that major software companies, Microsoft among them, have in recent years issued bare-bones manuals and help systems. (The help system for the Windows 95 operating system is a prime example.) Ummelen's answer, stated very briefly, is that declarative information is a highly useful component of computer documentation for a wide range of user and task situations.

One very difficult problem for a study of this kind is to formulate adequate definitions of "procedural" and "declarative" information. This problem is the focus of the first chapter. The chapter begins with a subtle analysis of the use of these terms in previous studies. Ummelen demonstrates how the meanings of these terms differ significantly from one study to the next and that these studies should be compared only with great care. She then proceeds to a detailed rationale for her own definitions. These definitions are based on both "text content" and "text form." In terms of text content, Ummelen defines *procedural information* as actions, conditions for actions, and results of actions. The actions may be highly specific (command syntax) or more general (the "computer task" or "real-world" level). So, for example, "Calculate the expenses," a real-world goal, is counted as procedural. In terms of text form, which encompasses both syntax and format, procedural information consists of steps in which the actions are expressed with an imperative verb. Declarative information consists of definitions, facts, explanations of system functions, and other "background" information and is presented in the indicative mood and often with modal verbs. Note that Ummelen's definition of *declarative* is quite distinct from that of Anderson's ACT* theory (1983). Given the protean nature of language, there are bound to be problems with any attempt to distinguish procedural and declarative information, and Ummelen acknowledges some here. Most notably, information that users will perceive as conveying an action can be expressed descriptively. Still, Ummelen's discussion of procedural and declarative information is itself a scholarly accomplishment, and her definitions provide a sound foundation for the study.

Journal of Business and Technical Communication, Vol. 11 No. 4 October 1997 502-505

The heart of the book consists of four chapters presenting highly detailed accounts of a chain of experiments, which can only be summarized here in generalized form. Ummelen divided her subjects (according to previous computer and spreadsheet experience) into three groups. Each group was asked to perform both a simple and a complex task with *QubeCalc*, a very limited but somewhat difficult character-based (non-GUI) spreadsheet program. The novice group, operating in "tutorial" mode, was told to work slowly so as to learn as much as possible about the software in preparation for a third task. The intermediate group, operating in "typical user" mode, was told to complete the two tasks as quickly as possible but to expect to perform a third task. The advanced group, operating in "reference" mode, was told simply to complete the two tasks as quickly as possible; they did not expect a further task. (Ummelen fully recognizes the confound here between levels of experience and goals; this decision was made to keep the experiments manageable in scope and because the goals she has assigned the subjects are typical for their levels of experience. Furthermore, a small control experiment suggests that the effects of the confound are slight.)

In three experiments the subjects were tested on five variants of the on-screen documentation that Ummelen prepared for the performance of these tasks. One variant consisted of only procedural information; the other four varied the presentation of both procedural and declarative information. The documentation was displayed to the subjects in blurry form—except for headings identifying the kinds of information that was blurred. Subjects clicked to get a clear view of the parts of the documentation they wanted to read, allowing Ummelen to maintain detailed logs (supplemented by videotapes) of their reading behavior. The subjects performed their *QubeCalc* tasks on a separate "logged" computer. This method of data collection does not seem obtrusive or highly unnatural, and yet it generates very rich data about user behavior. The method can be quite profitably employed in many kinds of information design studies.

Ummelen's findings will surprise at least some people in the documentation community. Over the broad range of conditions tested in this study, the subjects consistently chose to spend a significant amount of time (30%-40%) reading declarative information. Users want more than topic titles and steps.

The subjects' decision to invest significant time reading declarative information turns out to be a good one. The total time they spent reading documentation was no greater than that spent by the group that was given only procedural information. Their task performance (measured by task completion time, error rate, and efficiency in completing tasks) was equivalent to the procedures-only group. But reading the declarative information provided significant benefits: It led to faster performance in a delayed task, more factual knowledge of the software (as measured by a questionnaire), and better task transfer (as measured by a reasoning test). We can presume, therefore, that the presence of declarative knowledge in procedures enriches the subjects' schema for the software. These findings, Ummelen notes, accord with the theorizing and data of Kieras and Bovair (1984).

To design this study, Ummelen had to depart somewhat from conditions of real-world computer use. First, whereas real-world users formulate their own goals and decompose these goals into tasks and subtasks, Ummelen's subjects were given printed material explaining the general goal of each task and the subtasks making up the task. Inevitably, there were some unintended instructional benefits from reading this material, and these benefits could have affected the results of the study.

Second, Ummelen's subjects did not work with full-size commercial documentation but rather with a small number of procedures adapted from the very brief *QubeCalc* manual. If these subjects had been working with a full-size user's guide or help system, they would have expended significant effort searching for the needed procedures with such access devices as the table of contents and index. Ummelen's subjects had a relatively minor search task.

Finally (judging from the sample pages of the electronic manual provided in the appendix), the procedures Ummelen wrote depart somewhat from standard procedures. This difference is probably due to her need to maintain a greater-than-normal level of separation between procedural and declarative information.

We would certainly like to know about the choices users make and the consequences of these choices when they are working with standard procedures. In particular, a hot issue in the documentation community is the value of including an introductory paragraph between the topic title and the steps. Under what circumstances does the topic title tell the user enough so that the user is comfortable proceeding directly to the steps? Under what circumstances do users desire and need the introductory information? Ummelen's study does not bear directly on this issue.

Clearly, much work needs to be done before we have more complete answers to this and many other complex documentation issues. But this study—the product of a huge amount of work—gives us the best data so far as well as consistently sophisticated, insightful background and interpretation. A brief account of the first phase of this project is readily available (Ummelen 1996), but those with a serious interest in documentation will want to read the book.

REFERENCES

Anderson, John. The Architecture of Cognition. Cambridge, MA: Harvard University Press, 1983.

Kieras, David E., and Susan Bovair. "The Role of a Mental Model in Learning to Operate a Device." *Cognitive Science* 8 (1984): 255-73.

Ummelen, Nicole?]. "The Selection and Use of Procedural and Declarative Information in Software Manuals." *Journal of Technical Writing and Communication* 26 (1996): 475-88.