

# John Carroll's *The Nurnberg Funnel* and Minimalist Documentation

DAVID K. FARKAS AND THOMAS R. WILLIAMS

**The Nurnberg Funnel: Designing Minimalist Instruction for Practical Computer Skill**—John M. Carroll, MIT Press, Cambridge, MA 1990.

*Abstract*—In *The Nurnberg Funnel*, John Carroll reviews and reformulates his research on minimalism, a well-known approach to both print and online software documentation in which explicit instruction is severely reduced and users learn through a predominantly exploratory process. Although *The Nurnberg Funnel* is a stimulating and valuable book, Carroll fails to make a compelling case for minimalism as a broadly applicable alternative to the contemporary multi-component documentation set.

JOHN M. CARROLL, a social scientist at IBM's Watson Research Center, has conducted a very active research program in the areas of computer documentation and interface design for most of the 1980s. Among his influential publications are "The Minimal Manual" [1], "The Paradox of the Active User" [2], and "LisaLearning" [3]. A large portion of his work has concerned "minimalist" documentation, and this term has become widely known in the computer documentation community, though it is sometimes misunderstood simply as a commitment to brevity.

*The Nurnberg Funnel* has afforded Carroll the opportunity to sum up and reformulate his research on minimalism, and to describe some of his most recent projects. (The name of the book, incidentally, comes from the legendary Funnel of Nurnberg, through which knowledge could simply be poured into the brain of the learner.) At the same time the book affords the documentation community the opportunity to examine minimalism and the case Carroll makes for minimalist instruction.

This review is part of such an effort. We approach Carroll's work on minimalism both from the perspective of contemporary professional practice (Farkas) and instructional theory (Williams). The book, we find, fails to make a convincing case for minimalism as a design alternative.

Carroll originally developed the minimalist model in opposition to what he calls the "systems approach" to documentation, and in *The Nurnberg Funnel* he presents minimalism in direct contrast to the systems approach. The systems approach, however, is an outmoded and largely abandoned documentation model and is itself based on the largely aban-

doned psychological paradigm of behaviorism. Systems approach manuals are terrible. They are characterized, Carroll tells us, by extreme length and the decomposition of tasks into largely meaningless sub-tasks. We miss, therefore, a clear and direct comparison of minimalism to current and more functional documentation. Because systems approach manuals were used as controls in most of Carroll's empirical studies, the dramatic results often achieved by minimalist documentation are not persuasive. Finally, we have significant reservations about the efficacy of several components of minimalism—in particular, exploratory learning, which is probably minimalism's central and most distinctive component. At the same time, *The Nurnberg Funnel* is a rich book full of stimulating and useful commentary, and it takes a position that everyone in the documentation community should be familiar with.

## WHAT IS MINIMALISM?

The premise behind minimalism is that people learning to use computer software are impatient, mentally active, and curious. They want to begin right away getting their work done; they want to exercise their problem-solving abilities; and they are apt to utterly reject or diverge from highly constraining instruction such as tutorials. Training material, therefore, must not impede the natural impulses of computer users, as systems approach documentation does. It should be as brief as possible, support the accomplishment of real work, help learners recognize and recover from errors, and, when possible, permit non-sequential reading. Such documentation cannot be generated mechanically from a theory of instruction but requires careful attention to the needs and behavior of the intended users of the software and reiterative testing of the design.

This part of minimalism contains much that is not highly controversial; it contrasts more with systems approach documentation than with the contemporary documentation set. More controversial is the idea of exploratory learning, which entails radically cutting explanatory and procedural information so that learners may enjoy the challenge of exploring the software and mastering it through trial and error.

*The Nurnberg Funnel* presents minimalism as nine principles and applies these principles to a discussion of three major research projects, each centered on an innovative minimalist design. These designs are Guided Exploration, the Minimal Manual, and the Training Wheels Interface. But before we can productively consider these nine principles and Carroll's minimalist designs, we must examine Carroll's

D. K. Farkas and T. Williams are with the Department of Technical Communication in the College of Engineering, University of Washington, Seattle, WA.

IEEE Log Number 9040069.

account of the systems approach and contrast it to what we see as contemporary professional practice.

#### THE SYSTEMS APPROACH

Carroll refers to the systems approach to instruction throughout *The Nurnberg Funnel*, generally citing the 1979 edition of *The Principles of Instructional Design*, by R. M. Gagne and L. J. Briggs. For his actual examples of systems approach documentation, Carroll relies very heavily on an early IBM Displaywriter manual and on the print and online documentation for the Apple Lisa microcomputer.

The systems approach, Carroll tells us, consists of "the hierarchical decomposition of learning objectives" (p. 7). "Each objective is incorporated into a systematic lesson structure incorporating specific 'events' of instruction: gaining the learner's attention, informing the learner of the lesson's objective, stimulating the recall of prior learning, presenting stimuli with distinctive features, guiding learning, eliciting performance, providing information feedback, assessing performance, and enhancing retention and learning transfer" (p. 79).

In practice, notes Carroll, the systems approach results in horrendous documentation. The documentation is not task oriented, and typically little or no attempt is made to anticipate and deal with user errors. Learners are required to master skills "bottom up, by step-by-step drill and practice" (p. 7). Because the individual lessons are trivial, learners fail to see meaningful goals in the procedures they perform, impeding learning and squelching motivation (pp. 55, 75, 106). One of Carroll's examples illustrates the decomposition of tasks into minute and largely meaningless subtasks. When learning to use the Displaywriter's sign-on menu, users are taught the separate skills of cursor movement in menus, cursor movement in a menu field, typing characters, and using the enter key. Along with its other failures, such documentation is necessarily long and painfully slow-paced. The user of the Displaywriter manual does not even begin to create a document until reaching page 70.

It is not surprising, therefore, that Carroll, in his early studies, saw his research subjects rebel against such documentation and elect to teach themselves by experimenting with the system. In response, he developed a diametric alternative.

#### HOW PREVALENT IS THE SYSTEMS APPROACH?

The question arises, however, whether, as Carroll asserts, the systems approach is indeed the prevailing documentation model. Such manuals, to be sure, still exist, and people no doubt are still writing them. We have in fact noticed a book that propounds something very like the systems approach for the writing of tutorials [4]. There are, however, innumerable software applications, especially in the microcomputer and workstation worlds, whose documentation does not follow the systems approach. Furthermore, we have on our shelves various tutorials and other manuals for well-known hardware and software products with copyright dates of 1983 and 1984 (when Carroll's work on minimalism was in its early stages) that resemble contemporary documentation much more than

they do systems approach documentation. Although the contrast between minimalism and the systems approach is clean and direct, we cannot meaningfully assess minimalism as a design option without examining it in the context of contemporary documentation.

#### THE CONTEMPORARY DOCUMENTATION SET

Most software products, especially applications, come with several pieces of print and online documentation. Most relevant here are two fundamental pieces—the tutorial and user's guide—and one occasional piece—the quick start manual.

##### *The Tutorial*

Carroll uses the terms "training manual" and sometimes "self-instruction manual" to describe systems approach manuals. These manuals, however, in their intended function are much akin to what we know as tutorials. Both systems approach training manuals and tutorials try to recreate the experience of highly supportive, one-on-one instruction. Both are typically slow-paced and are intended in large part for the novice user or less aggressive learner. Both require users to trade off the opportunity to begin immediately on their own work in return for carefully controlled instruction—and so one of Carroll's minimalist principles is not met by these manuals.

But in contrast to systems approach manuals, the tasks that learners perform in contemporary tutorials are normally not hierarchically decomposed into meaningless units that are taught bottom up via drill and practice. Nor are they egregiously wordy. Typically, tutorials are made up of realistic task sequences, focused on relevant goals broadly similar to what users will be trying to accomplish with the product. As an example, the *WordPerfect Workbook* (Version 5.0) consists of a series of meaningful word processing tasks ("Underlining a Word"; "Setting a Tab Stop"), generally taught by means of a short explanatory paragraph, a series of numbered steps, and one or two screen representations. [5]

##### *The User's Guide*

Computer users, of course, have very different needs and abilities. In some instances, successfully completing a tutorial fully taxes the individual's problem-solving abilities and brings considerable satisfaction. These users, if given step-by-step instructional material other than a totally dysfunctional systems approach manual, will be grateful. On the other hand, it is well known that many users bypass even the best print and online tutorials. As Carroll rightly observes, the impulse to start getting one's actual work done immediately and the impulse to exercise one's problem-solving ability are strong indeed.

The user's guide is well suited for this class of user. It typically presents the complete functionality of the product in a logical hierarchy of task-oriented procedures. Individual procedures are something like recipes; there is a brief conceptual and descriptive overview, followed by numbered steps. User's guides support the user in accomplishing actual work, and they provide many opportunities for users to exercise their problem-solving capabilities. Users of these

manuals must map the myriad tasks presented in the manual to their own particular goals, and they very often have goals that are not precisely supported by the procedures in the manual.

The differences between a minimal manual and a user's guide, while certainly significant, are not overwhelming. Both are essentially task-oriented, procedural pieces. The main differences are that the minimalist manual is more committed to brevity, is more focused on error recognition and recovery, and—most important—is committed to the instructional strategy of exploratory learning with the product rather than learning supported by a full set of explicit procedures and conceptual overviews encompassing the product's full functionality. Learners fleeing systems approach training manuals and learners who want to bypass tutorials may well be as satisfied with a user's guide as with a minimal manual.

#### *The Quick Start Manual*

The contemporary documentation set does, in fact, include a piece specifically designed for exploratory learning. It is very brief, consisting only of those few things that an impatient user must know about the product before jumping right in. It is often termed a "quick start manual," and it is seen with some frequency. It is sometimes entitled "A Manual for People Who Don't Read Manuals," or it appears simply as a section of another manual, bearing a heading such as this: "If you read nothing else . . ."

The Quick Start Manual, however, is generally directed to the highly experienced, hacker-type computer user. The assumption, no doubt, is that an active, problem-solving frame of mind, impatience, and curiosity do not qualify the novice to forego more complete documentation.

The contemporary documentation set is diverse and includes command references, quick reference cards, online help, demos and tours, encyclopedic references, and more. These pieces serve different purposes and embody different assumptions about users and different instructional strategies. Most of Carroll's minimalist principles can be found in one way or another among these pieces. Unfortunately, except for brief allusions to reference manuals, Carroll never relates minimalism to the full range of contemporary documentation, and his book's value is thereby diminished.

#### MINIMALIST DESIGNS

We can now consider the three major minimalist designs and some lesser ones. These designs and the research projects centered on them embody much of Carroll's work on minimalism and are described in detail in *The Nurnberg Funnel*.

#### *Guided Exploration*

Guided Exploration, Carroll's first minimalist design, consists of a set of 25 separate cards, each providing minimalist instruction for a very basic function of the IBM Displaywriter. The cards support exploratory learning and contain "hints" rather than complete specifications for any procedure. The cards also contain considerable information for error recognition and recovery—that is, the cards anticipate

likely user errors, help users recognize when these errors have been committed, and show them how to recover. Research subjects using the Guided Exploration documentation attained greater mastery of Displaywriter than those using systems approach documentation; some of the Guided Exploration subjects, however, expressed discomfort with the unstructured nature of the learning materials, leading Carroll to develop the Minimal Manual.

#### *The Minimal Manual*

The Minimal Manual consists of bound pages, and so the reader is presented with at least an implicit reading order. Expository material of all kind is slashed drastically so that even with the addition of extensive error recovery material and exploration-type exercises, the average chapter length is three pages—one-fifth the average length of the systems approach chapters. Relatively meaningless chapter titles used in the Displaywriter training manual, such as "Using the Display Information While Viewing a Document," were eliminated in favor of task-oriented chapters such as "Printing Something on Paper." In empirical tests, subjects using the Minimal Manual exhibited both greater mastery and faster learning than did subjects using the systems approach manuals.

#### *The Training Wheels Interface*

In the next project, Carroll's goal was to greatly reduce the penalty entailed in exploratory learning. To do so, an alternative Displaywriter interface was developed that blocked user access to functions that were too complex for beginners and which had often confused and sometimes trapped subjects of previous studies. A series of experiments were conducted in which subjects used both the Training Wheels Interface and the complete system, supported by both systems approach and minimal manuals, in various combinations. Results were positive: the Training Wheels Interface reduced error recovery time and apparently helped learners form a better mental model of the system.

The Training Wheels Interface seems like an excellent instructional strategy; it is especially valuable in the context of exploratory learning, because so much error occurs. But the idea of blocking incorrect or nonproductive options can be used in online tutorials, and is closely related to the idea of staged interfaces, in which the more complex functions of a software product are hidden from the novice user. In one sense, the Training Wheels Interface is a move away from rigorous minimalism, since users cannot get their own work done if that work happens to require the use of a blocked function.

#### *Other Projects*

Carroll also offers relatively brief accounts of several projects that are preliminary or inconclusive in nature. The unifying idea behind these projects is to develop interfaces with "task intelligence"—that is, to endow systems with special understanding of user goals and actions so as to better support minimalist learning. Collectively, they show us that

Carroll is now applying his minimalist perspective to the potential uses of artificial intelligence in computer training.

The Scenario Machine is an extension of the Training Wheels Interface in which, for experimental purposes, all system functions but one are blocked. This enables the system to provide context-sensitive feedback to the user, which was not possible in the Training Wheels Interface. In the Smart Help project an unseen human expert plays the part of a help system and provides unusually apt responses to the user's queries and problems. Task Mapper is a preliminary project consisting of a specialized interface management tool that lends task-oriented guidance to users performing tasks involving multiple windows. View Matcher is an advanced learning environment for the Smalltalk programming language. It supports exploratory learning by providing sample Smalltalk applications (e.g. a tic tac toe game that the learner can play against the computer) while a separate window shows how the code underlying the game changes with successive moves.

#### NINE MINIMALIST PRINCIPLES

We now have the perspective to focus on the nine principles that make up the minimalist model and underlie Carroll's designs.

##### A. Getting Started Fast

As a goal, letting the computer user start using the software quickly is noncontroversial. The question is whether we must make damaging sacrifices to achieve this goal. Carroll enjoys some special advantages in cutting introductory material. Because his research subjects sit down at fully prepared machines, they do not need information about hardware compatibility and software installation. Because the Guided Exploration and the Minimal Manual are one-piece documentation sets, strategic information on when to use a tutorial, user's guide, command reference, and so on, does not appear. For the most part, however, documentation writers should be able to get users started reasonably quickly.

Brevity—which Carroll takes up under the rubric of getting started fast—is also noncontroversial as a goal. It has been an ideal of rhetoricians for many centuries. The question is how much can we cut. The striking departure in minimalism is that Carroll is willing to drastically cut explanatory information and procedural information and let the user learn largely from exploration.

##### B. Training on Real Tasks

Carroll recognizes fully the tradeoff between letting users begin doing real work and providing simulated tasks in the interest of controlling the learning experience. His position is that the motivational benefits warrant letting users do their own work, even when designer-specified tasks “contribute more to an ideal instructional curriculum” (p. 80). The tutorial/user's guide combination is another approach: one component uses simulated tasks and another permits the accomplishment of real work.

##### C. Reading in Any Order (Modularity)

Modularity is a complex and generally neglected issue, and Carroll's several discussions of it in *The Nurnberg Funnel* are most welcome. Increasing the degree of modularity in a document is most beneficial, for modular structure enables users to make sense out of the sections directly related to their goals without having to backtrack through earlier, less relevant sections. Documentation writers, therefore, often work hard to achieve modularity, seeking to create manuals in which, “You can just read the sections about the tasks you want to do” [6].

Modularity is especially important when users will be engaging in exploratory learning, leading Carroll to design for complete modularity in the Guided Exploration project and a high degree of modularity in the Minimal Manual. Unfortunately, however, achieving modularity is not always easy, especially when there are significant dependencies inherent in the software itself. In such cases, the designer must lengthen each module so as to include extensive pre-requisite information or must rely on extensive cross referencing—two tactics that are antithetical to minimalism. Of course, a possible minimalist alternative would be to prepare documentation that ignores these dependencies and thereby let the learner try to figure them out through an exploratory learning process. This experience, however, might prove unnecessarily taxing for many individuals. All told, the difficulty in achieving modularity in the context of the full minimalist model casts doubt on the range of products for which minimalist manuals can be written.

##### D. Exploiting Prior Knowledge

Meaningful learning is virtually impossible without exploiting prior knowledge, and all intelligently designed documentation attempts to do this. One of Carroll's important contributions over the years has been his sensitive account of user behavior and patterns of error, and his sophisticated commentary on the difficulty of successfully exploiting prior knowledge [2], [3]. Carroll strives to exploit the user's prior knowledge in all of his designs, and his treatment of this issue in *The Nurnberg Funnel* is cogent and valuable.

##### E. Coordinating System and Training

Novice users of computer systems have difficulty coordinating the system with training material. They do not notice discrepancies between what appears on the screen and what is shown or written in the manual. Or, they notice but respond inappropriately. Or, they treat expository information in the manual as procedures to be followed. Certain difficulties are specific to online tutorials—in particular the tendency of users to confuse the instructions and controls of the tutorial with those of the product's working interface.

Coordinating the system and training (or helping the learner to do so) is a challenge to all documentation designers. The minimalist strategy is, in large part, to keep the user's attention on the screen as much as possible, by drastically limiting the size of the manual, by using an inferential prose style, and by other means. This strategy will reduce the kinds

of user errors that arise from the user's failure to properly coordinate either the manual and the product or else an online tutorial and the product. It is potentially a real strength of minimalism.

But if we intend to allow users to rely largely on the interface itself to learn a piece of software, we must consider the difficulty of any particular interface for a particular class of users. If the product domain is sufficiently familiar to the user and the interface sufficiently suggestive, people can learn a piece of software without *any* form of documentation or any special training-type interface. But a great many software products embody unfamiliar and subtle concepts and have interfaces that do not suggest the function of the software to the user. Consequently, the prospect of learning without any documentation or through an exploratory training process becomes slimmer. In Carroll's major empirical studies, the research subjects perform very simple or relatively simple tasks on products designed for general office work. To assess minimalism adequately as a design alternative, we need more attention to the question of the relationship between minimalism and the nature of the software and empirical studies of users performing more sophisticated tasks on more complex systems.

#### F. Supporting Error Recognition and Recovery

It is indeed desirable to support error recognition and recovery. In a sense, the frequent use of screen representations in contemporary documentation assists in this. Users can confirm that the actions they take produce the proper result. Also, it is fairly common for manuals to anticipate user errors and to provide correctives for errors, and Carroll's earlier publications no doubt further this very beneficial practice.

Supporting error recognition and recovery is especially important in Carroll's designs, since users will inevitably make, and indeed are expected to make, many errors. Careful developmental testing of user behavior is required to determine what error recognition and recovery information to provide or, in the case of the Training Wheels Interface, what functions to block. In the case of Guided Exploration and the Minimal Manual, some of the savings in page count achieved by the reduction of explanatory material is lost by the extensive amount of error recognition and recovery information. The power of the Training Wheels Interface is that error correction is instant and without penalty, and is accomplished independently of the documentation.

#### G. Using the Situation

This principle is closely tied to exploratory learning and to error recognition and recovery. The idea is to provide the user with many opportunities to reason about how the system works. For this to occur, however, users must be able to learn from their errors as well as from their successes. Effective support for error recognition and recovery will prevent users from getting into complex error states that will cause confusion and frustration and thereby prevent learning from taking place.

#### H. Developing Optimal Training Designs

This principle differs from the others in that it pertains to the *process* of preparing the documentation rather than to the product. The premise is that neither minimalism nor any other instructional model leads directly and deductively to a design. More broadly, it is mistaken to assume that scientific knowledge leads directly to innovative technologies or designs in any domain; innovations often arise independent of and prior to the theoretical knowledge that accounts for them.

The implication of this idea is that each minimalist design—and indeed all designs in the realm of human-computer interaction—must be developed empirically through careful study of user needs and behaviors in specific, realistic domains, and through reiterative testing of preliminary designs.

Carroll's view that design checklists or guidelines are inherently sterile and mechanical seems extreme—guidelines can serve as simple heuristics to spur creative thinking or, at least, to ensure that relevant perspectives on a problem have not been forgotten. But there is little else to question concerning this minimalist principle, and the commentary on design and design methodology that Carroll offers throughout the book is excellent and will give technical communicators a broader and more sophisticated understanding of their work.

#### I. Reasoning and Improvising

The principle of Reasoning and Improvising includes exploratory learning, probably the most distinctive, far-reaching, and controversial aspect of minimalism. It also includes the closely related concept of inferential learning.

*Exploratory Learning:* Do the advantages of exploratory learning outweigh the value of appropriate explanatory information followed by explicit, task-oriented procedures? One benefit of exploratory learning is the motivation that comes from the challenge, and Carroll compares exploratory learning to people's enthusiasm for computer games. On the other hand, another powerful motivator is success, achievable with the user's guide when users get their work done; and, in a tutorial, when users learn what they will need to turn confidently to their own work.

Users will often induce the principles of a software program through exploration, but, as Carroll notes, this effort may not succeed. In particular, users can readily induce only partially correct mental models that will not serve them in good stead later on. One can well argue that appropriately written conceptual overviews and explicit procedures will prove more reliable in helping users develop a conceptual understanding of the product. Furthermore, the process of learning through exploration requires extra mental effort that may not always be acceptable to users. But again, in favor of exploratory learning is the argument that learners are less prone to make certain kinds of errors when they attend more to the software itself and do less moving between the working interface and either print or online instructional materials.

Carroll's empirical findings show that minimalist manuals usually result in faster learning than do systems approach manuals. But impatient users, while they want to read as little documentation as possible, may prefer to be told exactly what

to do rather than to learn by trial and error, even when this process is assiduously supported by the documentation or interface.

An important issue is retention. While exploratory learning promotes retention [7], users do not always seek to retain procedures that they do not intend to execute regularly. In such cases, following steps may well make more sense than expending the greater effort (and perhaps time) entailed in exploratory learning. If users do execute a set of steps frequently, they will begin to retain them in memory. But users should have a choice as to whether they are reading to retain or simply to execute procedures.

*Inferential Learning:* Inferential learning entails learning from information that is inexplicit or incomplete. Carroll endorses inferential learning as a means of promoting comprehension, retention, and active involvement in the learning process, and he cites two somewhat inconclusive studies in which inferentially written minimal manuals yielded results that were in some respects superior to skeletal minimal manuals.

The idea of an inference, however, is elusive and difficult to apply to instruction. As E. D. Hirsch points out, no discourse is truly explicit; speakers and writers always assume that audiences will exercise their powers of inference [8]. We suggest, in fact, that there is a continuum from normal discourse through intentionally inferential instructional prose to documentation designed for exploratory learning (where inferences become very large). Carroll offers no metric for the size of inferences; often the inferences he refers to seem like small ones, rather like stylistic devices. An added complication is that inferences vary, as Carroll observes, according to kind—that is, general to specific, analogy, and so on. Although inference is certainly an elusive concept and its effects difficult to measure, we do not doubt that inferentially written instructional prose can have benefits. But, we suggest that difficult design problems are apt to emerge as one moves away from small inferences and into the domain of exploratory learning.

#### CONCLUSION

John Carroll has written a rich and thought-provoking book. It is excellent in elucidating the complex relationship between the design process and basic science and offers many valuable insights about the motivations and behavior of computer users. It raises—though does not necessarily resolve—a variety of important issues, including modularity, learning from real vs. designer-specified tasks, and exploratory learning. The book also provides a fully adequate demonstration of the total dysfunctionality of the systems approach to computer instruction—and such a demonstration is needed in some quarters. Most important, *The Nurnberg Funnel* is now Carroll's most complete and most current argument for minimalism and includes descriptions of a very diverse and interesting set of minimalist designs.

On the other hand, a flexible, multi-component documenta-

tion set in which diverse learning styles are supported is a compelling alternative to minimalism. It will certainly be preferred by users who cannot master software through trial and error or who have no taste for such a process. It also steers us past what looks like a severe minimalist design dilemma: trying to achieve modularity and extreme brevity while explaining complex, interdependent procedures. Our reservations about minimalism do not, however, apply to the Training Wheels Interface; this is a very sound instructional technique whether it is implemented in a minimalist context or used in conjunction with a standard online tutorial.

Carroll makes brief mention of some successful commercial documentation projects based on minimalist principles, but considering that minimalism has been fairly well known for at least four years in a very fast-moving industry, one cannot say that the marketplace has spoken strongly for minimalism. We need to see the case for minimalism made, both in terms of conceptual arguments and empirical findings, against up-to-date tutorial instruction and the multi-component documentation set. Also, Carroll has not fully addressed, conceptually or experimentally, the question of the relationship between the difficulty and structure of the software and the effectiveness of minimalist instruction. Thus, *The Nurnberg Funnel*, despite its considerable interest and value, is not a fully satisfying book and leaves open the question of whether minimalism is a superior design option for a broad range of software products.

#### REFERENCES

- [1] Carroll, J. M., Smith-Kerker, P. A., Ford, J. R., and Mazur-Rimet, S. A., "The Minimal Manual," *Human-Computer Interaction*, vol. 3, pp. 123-153, 1987-88.
- [2] Carroll, J. M. and Rosson, M. B., "The Paradox of the Active User," in *Interfacing Thought: Cognitive Aspects of Human-Computer Interaction*, J. M. Carroll, Ed. Cambridge, MA: MIT Press/Bradford Books, 1987, pp. 80-111.
- [3] Carroll, J. M. and Mazur, S. A., "LisaLearning," *IEEE Computer*, pp. 35-49, Nov. 1986.
- [4] Crandall, J. A., *How to Write Tutorial Documentation*. Englewood Cliffs, NJ: Prentice-Hall, 1987.
- [5] *WordPerfect Workbook for IBM Personal Computers* (Version 5.0). Orem, UT: WordPerfect Corporation, 1989.
- [6] *Microsoft Excel User's Guide* (Version 1.0). Redmond, WA: Microsoft Corporation, 1985, p. xv.
- [7] Bruner, J. S., Oliver, R. R., and Greenfield, P. M., *Studies in Cognitive Growth*. New York: Wiley & Sons, 1966.
- [8] Hirsch, E. D., Jr., *The Philosophy of Composition*. Chicago: University of Chicago Press, 1977, p. 100.

**David Farkas and Tom Williams** teach in the Department of Technical Communication in the College of Engineering at the University of Washington. Farkas' teaching and research interests include computer documentation, document design, hypertext, and human-computer interaction. His article on encyclopedic reference manuals appeared in the May 1990 issue of *Technical Communication*.

Williams' teaching and research interests include learning and cognition, graphics, and production editing. His article on the applications of schema theory to problems in technical communication is forthcoming in a collection edited by Charles Sides for the Association of Teachers of Technical Writing.