Human-Computer Interaction
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/hhci20

The Minimal Manual

John M. Carroll , Penny L. Smith-Kerker , James R. Ford & Sandra A. Mazur-Rimetz
Published online: 11 Nov 2009.

HUMAN-COMPUTER INTERACTION, 1987-1988, Volume 3, pp. 123-153

The Minimal Manual

John M. Carroll
IBM Thomas J. Watson Research Center

Penny L. Smith-Kerker
IBM Austin Laboratory

James R. Ford
IBM Federal Systems Division

Sandra A. Mazur-Rimetz
Micro Control Systems

ABSTRACT

The Minimal Manual was designed to address difficulties people have with state-of-the-art self-instruction manuals in learning to use powerful computing devices. It is briefer; it helps learners to coordinate their attention between the system and the manual; it specifically trains error recognition and recovery; it better supports reference use after training. In two experiments, the Minimal Manual was shown to afford more efficient learning progress than an otherwise comparable, commercially developed self-instruction manual, and was superior in the specific areas predicted by its design.

This article was originally given at the 1984 annual meeting of the Psychonomics Society in San Antonio, Texas. The complete Minimal Manual is published as an appendix to Carroll, Smith-Kerker, Ford, and Mazur (1986), available from the first author.

Authors' present addresses: John M. Carroll, User Interface Institute, IBM Thomas J. Watson Research Center, Box 704, Yorktown Heights, NY 10598; Penny L. Smith-Kerker, Development Human Factors, IBM Austin Laboratory, 11400 Burnet Road, Austin, TX 78758; James R. Ford, IBM Federal Systems Division Headquarters, 6600 Rockledge Drive, Bethesda, MD 20817; Sandra A. Mazur-Rimetz, Micro Control Systems, 27 Hartford Turnpike, Vernon, CT 06606.

CONTENTS

1. LEARNING TO USE A WORD PROCESSOR

Learning to do something new is always a difficult undertaking. Trying to get someone else to engage in such learning is even more challenging. Perhaps it is the challenge that has maintained learning as a traditional focus of interest in both theoretical and applied psychology. Recently, concern with learning has intensified in the study of human-computer interaction. The principal reason for this is the rapid extension of computer use to people who are not programmers or other computer professionals. Large numbers of people are learning to use computer equipment, and many of them are having trouble.

To some extent this is not surprising. Computer applications and their user interfaces are often very complex. However, some of the problems people have in learning to use computers are caused by the materials provided to help them learn. Education in the computer industry has traditionally concerned itself with the training needs of computer professionals, who indeed may be willing to study comprehensive but very thick reference manuals, and to attend expensive and intensive classes. Many new users have little interest in computer science, programming, or electronics per se. Rather, they are concerned with preparing manuscripts, letters, memos, and other sorts of documents (Davis, 1984; Eason, 1976). Currently, the industry is concerned that overly comprehensive materials may exhaust the patience and the technical backgrounds of users (Davis, 1984; Scharer, 1983).

The materials provided to new users often conflict with the learning styles and strategies they adopt spontaneously. Standard self-instruction training manuals and on-line tutorials require the user to proceed step-by-step through sequences of drill and practice exercises (Uhler, 1981, 1984). However, our studies of such stand-alone education showed that deliberate step-by-step progress was frequently interrupted by episodes of self-initiated problem solving (Carroll & Mazur, 1986; Mack, Lewis, & Carroll, 1983). In their statements and actions, users expressed a preference for getting started immediately on recognizably coherent tasks, like typing a letter, in favor of pursuing drill and practice. They just ignored steps and sections of the training that seemed irrelevant to their task-oriented concerns. They seemed quite willing to rely on their own inferences, even though these were based only on hints gleaned from whatever system prompts and messages they noticed and on highly erratic references to the manual, and even though these inferences often led them to make mistakes.

This active orientation to learning is not what the designers of self-instruction training intended, and was in fact poorly supported by the training. Much of our prior work is a recounting of the many problems new users encountered as they tried to learn systems using these materials. In the work reported here, we empirically developed a training manual for the word-processing functions of a very widely used, stand-alone, commercial office information system. We very deliberately constructed a manual that differed from the commercially developed self-instruction training manual in specific ways (ways our prior work had suggested would be important; see Carroll, Mack, Lewis, Grischkowsky, & Robertson, 1985). We then tested learning performance in a simulated office environment (Experiment 1) and then, to gather more detailed learning data, in one-on-one close observation (Experiment 2).

2. A MINIMALIST TRAINING MODEL

Our strategy in training design is to accommodate, indeed to try to capitalize on, manifest learning styles and strategies. In particular, (a) we address the preference to get started immediately on real tasks by structuring training to accommodate this; (b) we address the preference to skip in reading by presenting less material to be read; and (c) we address errors, an important consequence of the active orientation to learning, by supporting error recognition and recovery.

2.1. Focus on Real Tasks and Activities

A major theme in standard instructional models, such as the "systematic approach" (Gagne & Briggs, 1979; see also Mager, 1975), is the hierarchical decomposition of learning objectives. There is no reason in principle why an analysis of learning objectives could not be coupled with a synthesis of those objectives into a program of realistic tasks for learners, but in fact the standard presentations of systematic instructional design do not provide examples of this. Rather, they focus on curricula for building skills from the bottom up, via step-by-step drill and practice. In our prior work we were repeatedly struck by the tenacity with which learners would try to accomplish a real task despite the step-by-step guidance of their training materials, and by their unhappiness and lack of success when they tried to use the materials (Carroll & Mazur, 1986; Mack et al., 1983). As one of our learners put it, "I want to do something, not learn how to do everything." When learners abandoned their own goals and followed the drills successfully, they sometimes had difficulty extracting any meaning from their efforts, as one person put it, "What did we do?"

New users of application systems are trying to use a tool, a tool that they believe will help them do their own work. They are not learning for learning's sake. They come to the learning task with an often considerable understanding of task-relevant concepts and the motivation to learn to use the tool in their work. Training should make it easy for them to use the knowledge they already have (e.g., a new word processor user might know what a blank line is but not when it is referred to as a "carrier return control character in the data stream"). To the greatest extent possible, training should involve real tasks: Learners should do a real calculation with their electronic spreadsheet, one with their own numbers. People learning messaging systems should be sending messages (or at least using a convincing simulation); they should not be merely preparing and then canceling messages. This is not instructional libertarianism, it is practical psychology: The most important factor in learning is learner motivation, but this is also the factor least amenable to extrinsic control via design. If learners want to undertake a particular activity, then letting them attempt to do so is perhaps the best design step we can take.

2.2. Slash the Verbiage

New users are not inclined to read training material. As one person we observed put it while flipping pages in a manual, "This is just information." Users seem to be more interested in action, in working on real tasks, than in reading. We have found that learners are very susceptible to plunging into a procedure as soon as it is mentioned (e.g., in a preview) or of trying to execute purely expository descriptions (e.g., reviews). Of course, executing a preview may alter the system state and therefore make it impossible to execute the previewed exercise. Learners also often skip over crucial material if it does not address their current task-oriented concerns, or skip around among several manuals composing their own ersatz instructional procedure (Carroll & Mazur, 1986; Mack et al., 1983).

Self-instruction designs often responded to these reading problems by adding supplemental control information (e.g., injunctions to "not do anything before reading everything"; an initial chapter on "how to use this book"). Such material is intended to keep the learner properly oriented to the self-instruction, but necessarily adds to the sheer bulk and complexity of the training material. This in turn can add to learners' reading problems: by providing an even more imposing manual they will be even more disinclined to read and by providing more separate information types to be differentiated and confused. We have explored the radical alternative of eliminating more control information to address the problem of skipping and misreading, and of presenting material once and as briefly as possible to make the training unimposing (see also Reder & Anderson, 1980).

2.3. Support Error Recognition and Recovery

A typical assumption in the design of self-instruction training manuals is that learners will not make errors. But when learners pursue their own goals, when they explore, or even when they mistype, training problems arise that the self-instruction materials do not address, for example making a diagnosis of what has happened and how to recover. Even if the system being learned includes invertible commands, or a general undo command, the user may need guidance to try these. Turning the system off to start over is a crude and typical approach new users take, but in all of the systems we have studied there are serious error tangles that arise as consequences of doing this (e.g., Carroll & Carrithers, 1984; Carroll & Mazur, 1986).

Users often fail to coordinate their attention between training material and system events. They may read an exercise step, carry it out, and then go on to the next, never checking whether the step worked properly. Or, a given step may trigger an episode of self-initiated exploration. In either case, the system state and the training state may become unsynchronized. Training should make it easy to check the coordination of the training and the system. It is probably too much to expect that all errors can be avoided; instead, errors are probably best regarded as an inevitable part of learning. Training materials must therefore explicitly support the recognition of and recovery from error both to make the materials robust with respect to user error and to train error recovery skills. Particularly, in self-initiated forays of exploration, errors may play a unique constructive role in facilitating the discovery of new knowledge (Piaget, 1985; Schank, Collins, & Hunter, 1986).

2.4. Guide Exploration

The Minimalist Training Model was first developed and tested in our prior work on guided exploration (Carroll et al., 1985). In that work, we designed a set of brief cards to replace a commercial self-instruction manual. The cards stressed real work by addressing user-relevant goals (e.g., typing something and quitting work) over purely system-relevant topics (e.g., practicing with the status line and menu control). They contained about an eighth as much verbiage as the manual. They stressed error recognition and recovery by graphically breaking down all procedural information into four meaningful components: goals, enabling hints, checkpoints and results, and error recoveries.

This design enhanced learning. Guided exploration learners spent less than half as much time in learning as did their self-instruction manual counterparts. During learning they spent less than a third as much time reading, committed half as many errors, and recovered more often from their errors. These differences persisted in a transfer of learning posttest, in which the guided exploration learners accomplished half again as much in half the time as did the learners who had been training via the self-instruction manual. Nevertheless, guided exploration learners sometimes voiced a desire for a more structured training tool; in particular, they asked for a manual. We decided to develop an experimental self-instruction manual to try to capitalize on the strengths of the Minimalist Training Model and the desire of learners to have a structured manual.

3. DESIGNING A MINIMAL MANUAL

The Minimalist Training Model is not intended to be applied automatically in a single design pass. We cannot simultaneously have the briefest manual and the one that includes the greatest amount of error recovery information. As in other aspects of user interface design, and design in general, Minimalist Training is developed iteratively: designed, empirically evaluated, and then redesigned (Dreyfuss, 1955).

The role played by empirical testing may be greater for Minimalist training than for standard self-instruction materials. It is typical instructional design practice to do some *development* testing (also called *formative* testing; see Gagne & Briggs, 1979, pp. 37–38), but this is done late in the design process and its typical outcome is the addition of further material, qualifications, cautions, and additional explanation. In Minimalist design, empirical testing must enter the process earlier, for the core domain tasks to be trained are identified empirically. Even later development testing is different: The remedy of choice for a problem should be to cut, not add (though this can be very difficult to do). The particular iterative process we employed in designing our Minimal Manual is that described by Carroll and Rosson (1985), which consists of three stages of qualitatively different empirical testing.

3.1. Design Analysis

In the first stage, we analyzed the training situation. We chose to focus on the domain of word processing and developed an empirical understanding of the core tasks new users were motivated to undertake, the ways in which training materials for them could be cut down, and the key user errors our design would have to address.

Focus on Real Tasks and Activities. The goal of involving the learner was paramount. The chapters of our manual clearly labeled topics of interest to learners, for example, "Printing Something on Paper" instead of "Menus, Messages, and Helps." Learners created their first document only seven pages into the Minimal Manual. In the commercial manual, the creation of a first document was delayed until page 70. In the Minimal Manual, the first document creation was a letter; in the commercial manual, it was a description of office document processing. We tried to exploit users' prior domain knowledge in introducing word-processing concepts; for example, text block moves were referred to as "cutting and pasting" to trade on the metaphor of physical operations on pieces of paper.

We tried to make the procedures of the Minimal Manual more open-ended than in standard self-instruction, to better resemble real work and to maintain learner motivation. Procedural details were deliberately specified incompletely to encourage learners to become more exploratory, and therefore, we hoped, more involved in the learning activity; for example, the function of the cursor step-keys was introduced with an invitation to "try them and see." Open-ended exercises, entitled "On Your Own," were placed at the end of each chapter. For example: "As you can see on page 4:3, more deletions, insertions, and replacements are suggested for the Smith Letter document. Practice your revision skills by trying some of these. When you have practiced enough, print out Smith Letter." Indeed, the amount of work suggested to learners by such exercises was equal to that presented by the manual's self-instruction procedures.

Figure 1 presents a page of the Minimal Manual illustrating some of these points. We discovered in our pilot work that learners quite often accidentally add lines to a document. They want to delete these "blank lines," but from the perspective of the system, this translates into deleting a "carrier return control character" (a goal no novice user would even imagine having). As Figure 1 shows, the Minimal Manual explicitly addresses this problem in the

Figure 1. **Page from the Minimal Manual illustrating techniques for presenting system functions as real and familiar tasks under the user's control.**

Topic 6: 2

DELETING BLANK LINES

The Displaywriter stores blank lines as carrier return CHARACTERS.

USE ↑ TO POSITION THE CURSOR AT THE BEGINNING OF THE SECOND LINE OF THE FIRST PARAGRAPH OF Smith Letter.

PRESS CARRIER RETURN ONCE.

You have inserted a blank line in the paragraph.

USE ↑ TO POSITION THE CURSOR AT THE BEGINNING OF THE BLANK LINE -- ALL THE WAY AT THE LEFT.

As you can see, a special highlighted carrier return character appears. This is the special character that was inserted when you originally pressed CARRIER RETURN.

PRESS THE DEL KEY.

WHEN THE DISPLAYWRITER PROMPTS YOU: Delete What?, PRESS ENTER.

The blank line disappears. You have deleted the special CARRIER RETURN character.

ON YOUR OWN

You can use these techniques to insert and then delete underlined and centered material. Experiment with deletion. When you are finished, END the Smith Letter document, and then print it out.

---

goal-vocabulary of the learner. Most of the topic chapters of the Minimal Manual include open-ended exercises in which the learner can use the system and the manual to plan and carry out some activity.

Slash the Verbiage. We tried to be ruthless about verbiage, constructing a bare-bones manual, in the end 45 pages or less than a quarter the length of the commercial training manual. We achieved this by eliminating all repetition, all previews, reviews, and practice exercises, the index, the "welcome to word processing" introduction, and the troubleshooting appendix. By streamlining the manual, we hoped that reptition would not be necessary, that learners who forgot how to print for their second document would just look back at what they had already done. By addressing tasks learners understood on the basis of their prior familiarity with the domain, we hoped to avoid the need for previews and reviews, as well as the problems that occur when learners execute them. We viewed the manual itself as guided practice, and eliminated further rote exercises in favor of the more realistic On Your Owns. Because we had never seen successful use of indexes and troubleshooting appendixes in training manuals, but had seen many errors triggered by their misuse, we decided that these too could be eliminated.

Chapters were organized to be brief (averaging less than three pages), so that learners could easily move from topic to topic. Task-oriented chapter headings (e.g., "Centering and Underlining") were employed so that the table of contents could itself serve as an effective index. All material not related to office work was eliminated or radically pared (the Welcome overview, the descriptions of the system status line, the details on the system hardware components, the chapter entitled "Using the Display Information While Viewing a Document," etc.). We also tried to simplify wording: The term *the system* was replaced by the proper name of the system; function key names (e.g., CTR) were replaced by more transparent referring terms (e.g., *the CTR key*); the terms *display* and *display module* were changed to *screen;* the term *keyboard module* was changed to *keyboard.*

Support Error Recognition and Error Recovery. We inventoried the principal errors of new users and included specific error recovery information to address these problems. For example, we had found that learners had trouble with the diskette name concept and often typed an incorrect diskette name when prompted, which had the effect of leaving the system hung up (i.e., prompting for a diskette that did not exist). The system function in fact provided a specific recovery procedure for this problem, but the commercially developed self-instruction manual failed to mention it (it was a compound keypress and may have been thought to be too complicated for new users, but it was also crucial for them). The Minimal Manual included the specific error recovery information for this error. Another error that was typical for learners was pressing the Cancel key without holding down the Code key (which is also illustrative of the general problem of misexecuting compound keypresses). Cancel is perhaps the best general error remedy the system offers, but has an entirely different meaning when used without the Code key, and one which leads to complex side effects. What complicates the error even more is that the recovery for pressing Cancel without holding the Code key is pressing Cancel *while holding* the Code key (which allows this error to tangle up with itself). Throughout the Minimal Manual the key was explicitly referred to as "Code + Cancel," to stress the correct key combination both as an error prevention and as an error recovery—and we referred to the combination frequently (to remind learners of its general use in error recovery).

We also tried to help users avoid making errors, in particular by helping them to keep the manual and the system properly coordinated. Instead of merely saying that the system would prompt for a document name, and that they should type the name and press ENTER, the Minimal Manual asked "Can you find this prompt on the display?: 'Type document name; press ENTER.' " We specified procedures incompletely in the manual when required information could be found on the display. Instead of specifying whether a given diskette should be loaded in the right or left disk drive, we left it to the learner to consult the system prompts. The manual intentionally included very few pictures of displayed system screens, to impel learners to coordinate the actual system display with description in the manual. Finally, indirect references were used as soon and as much as possible. Once learners had had some practice selecting document creation in the Typing Tasks menu via direct prompting ("Type *a* and press ENTER"), the manual switched to an indirect mode of reference ("Choose the item in the Typing Tasks menu to create a document").

Figure 2 presents another page from the Minimal Manual illustrating some of these points. The interrogative prompt "Can you find this prompt on the screen?" helps to coordinate the learner's attention between the system and training: If indeed the learner can find that particular prompt at that point in the training, then the system and the training are quite likely to be synchronized. Turning things around, if the learner cannot find the prompt, then an error condition is quite likely. The triangular symbols indicate error recovery information.

3.2. Subskill Testing

After the design analysis, particular design elements were empirically tested in qualitative detail. We observed typical users performing typical tasks. Our goal was to gather detailed information that could be directed at redesign. This work (carried out by Caroline Carrithers and the first author) involved about 20 learners for sessions of between 2 hr and 8 hr each. In many cases, we were gratified to find that our design worked: Problems we had inventoried for learners using the commercial manual had been eased. For example, we had seen many learners suffer from the complications of typing an unprintable character into a text file (the system suspends the print job until a special request is made by the user). A particularly difficult aspect of the error is that

Figure 2. **Page from the Minimal Manual illustrating techniques for helping to coordinate user attention and to support error recovery.**

Topic 2: 1

Topic 2. Typing Something.

In the terminology of the Displaywriter you will be "creating a document" -- that is, typing a brief letter. You will first name the document, as you might make up a name for a baby **BEFORE** it is actually born. Then you will assign the document to a work diskette -- this is where the document will be stored by the Displaywriter. And then, finally, you will type the document at the keyboard, and see the text appear on the screen -- your electronic typing paper!

Note: If you turned off the power at the work station between Topic 1 and Topic 2, turn it back on and re-insert the Vol. 1 program diskette.

TASK SELECTION

In the TASK SELECTION menu you choose the general kind of work that you want to do. To type (create) a document, choose TYPING TASKS.

Can you find this prompt on the screen?:
Type ID letter to choose ITEM; press ENTER.

TYPE AN a, THE ID LETTER FOR TYPING TASKS, AND THEN PRESS THE ENTER KEY.

▶ If you typed the wrong ID letter, just press BKSP (backspace) to remove the incorrect letter.

▶ If you typed the wrong letter **AND** pressed ENTER too, hold down the CODE key and press CANCL enough times to return to the Task Selection menu to start over. CANCL and REQST are on the same key. If you don't hold down CODE when you press CANCL, you will get REQST (Request) instead of CANCL. You can correct this by trying CANCL again, this time while holding down CODE.

the feedback provided by the system is obscure; hence, learners usually do not even recognize that they have committed an error. Therefore, we included specific information to help recognize this error and further information for recovery. And indeed, the learners we observed were successful with the error.

In other cases, our initial testing uncovered problems with the design. Three problematic subskills were document naming, canceling, and turning the system off as an error recovery. Learners still had trouble with the concept of document name despite our attempt to better clarify it in the manual. We ended up by having to add a substantial amount of additional information on the concept (almost a half page). However, this was not a matter of reinstating material we had originally purged. A specific problem in naming was that people found it unintuitive that a document they had not yet created had to have a name. We drew upon the metaphor of "naming a baby before it is actually born." We also included more task-related motivation, expanding the bare statement "You must give a NAME to the document you are going to type" to "You must NAME documents (letters, memos) as you would label a file folder—so you can get these documents back to work on later."

Similarly, we were persuaded to embellish our diagnosis/recovery information for the Code + Cancel error. Learners still had a lot of trouble coordinating the keypress and making sense of the consequences when they failed to do so. We added the following material to the first two references to the operation: "Cancel and Request are on the same key. If you don't hold down the Code key when you press Cancel, you will get Request instead of Cancel. You can correct this by trying Cancel again, this time while holding down the Code key."

We also observed a problem with turning the system off as an error recovery. We explicitly made use of this recovery method with the suggestion "Turn the system off, but be sure you first REMOVE ALL DISKETTES FROM THE DISKETTE UNIT." It seemed that learners were performing this as they read it, and therefore switching off before they read the capitalized condition (which had complicated side effects). We substituted this: "REMOVE ALL DISKETTES FROM THE DISKETTE UNIT TO AVOID DAMAGING THEM and then turn the system off." We also decided to make this recovery just a bit less attractive to learners by substituting "You must now start all over again, by reloading your programs from the program diskettes" for "You can now start fresh from the beginning." We found that turning the system off is best used as a last resort. Even if it is executed errorlessly, it disrupts work.

We had originally hoped to introduce document revision as an On Your Own exercise, to have users "discover" revision instead of being taught about it. This did not work out; the subskill had to be further decomposed in order to be reliably executed by learners. These are only examples; there were many specific subskill problems we detailed and for which we then redesigned. (For further discussion of the design of the Minimal Manual, see Carroll, 1984, 1985. The Minimal Manual we developed is reproduced as an appendix to Carroll, Smith-Kerker, Ford, & Mazur, 1986).

3.3. Criterion Testing

Although reiterative subskill testing guarantees a sort of local optimization of the design, it is not directed at providing an objective benchmark assessment of the final success of the design. It is useful in the end to know just how good a design really is (e.g., relative to other contrasting designs or relative to particular usability goals). How good a training manual design did we end up with? We performed two laboratory studies to test and to better understand the empirical efficacy of the Minimal Manual design. The first of these focused on performance in 3 days of simulated office work experience. The second was somewhat more analytical in trying to expose the underpinnings of the performance effects, but also more limited in focusing on only a daylong learning experience. Sections 4 and 5 detail these experiments and their results.

4. EXPERIMENT 1

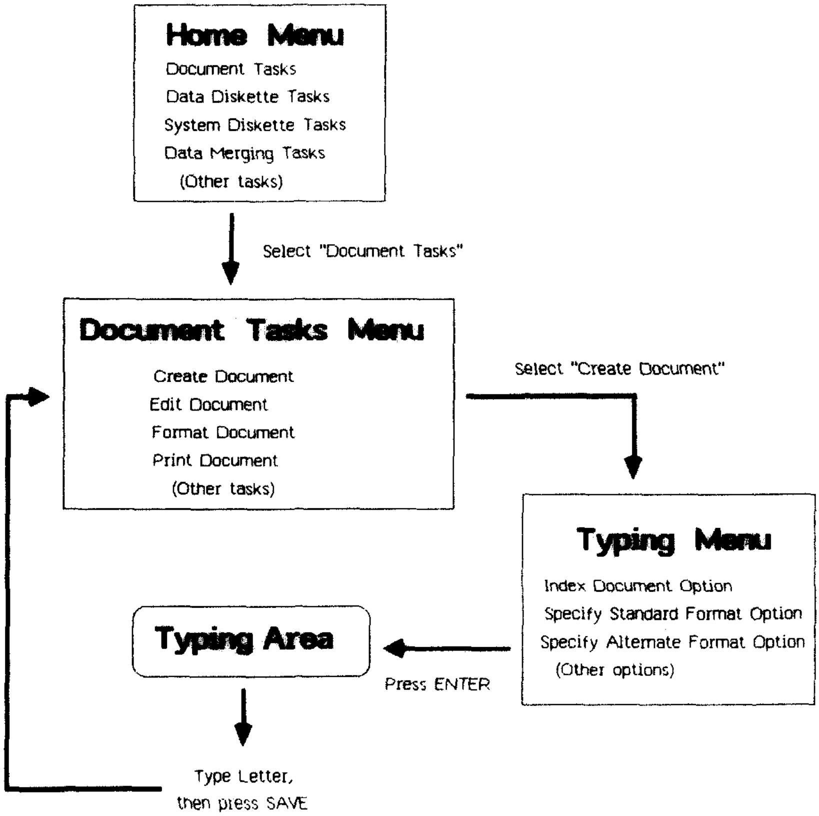
The first experiment was carried out by Smith-Kerker and Ford at the IBM development laboratory in Austin, Texas. The primary purpose of this experiment was to contrast a commercially developed standard self-instruction manual (SS) with the experimentally developed Minimal Manual (MM) in an officelike environment.

4.1. Method

Design. The design of this experiment was a between-subjects contrast of the independent variable of manual (MM or SS). Experimental sessions lasted up to 3 continuous working days, 8 hr per day. A reiterating study-test procedure was employed. Subjects were asked to learn on their own and to perform periodic performance tests.

The System. The system we studied is a commercially available menu-based office information system. Our study focused on its word-processing function, which in practice is the function initially attempted by clerical users. Figure 3 schematizes the flow of control for the document creation task in this system, the most basic and one of the most typically engaged word-processing tasks.

Figure 3. **Flow of system control in the word processor for the "create a document" task.**



The system's function is initialized by loading a system diskette. As indicated in Figure 3, this brings up a Home menu, which presents a variety of selection alternatives to the user. To create a document, the user selects "Document Tasks." This brings up the Document Tasks menu; the user selects "Create Document." This brings up the Typing menu, which presents options (e.g., formatting options) for the document to be created. The user then presses Enter, to go to the Typing Area. In the Typing Area, the user can input text (using typewriting keyboard functions and keystroke commands, like Center and Underscore).

The keystroke command Save writes the user's document onto a data diskette and returns control to the Document Tasks menu, from which the user can create further documents, or edit or print an existent document. For example, printing a document is accomplished by selecting "Print Document" in the Document Tasks menu, specifying options in a Print menu (analogous to the Typing menu), and then pressing Enter to send the document to a printing device.

Subjects. Nineteen subjects participated in the experiment, 10 in the MM condition and 9 in the SS condition. A temporary agency screened participants for two qualities: (a) We wanted people who were experienced with routine office work, typing in particular; and (b) we wanted people with little prior computer experience, defined in the case as less than 3 months of overall word-processing experience and no experience with the particular system we studied. In fact, 10 of the subjects we obtained had less than 3 months of prior experience with word processing (4 in SS, 6 in MM), and the remaining 9 had more than 3 months' experience. However, we were able to ensure that no subject had prior experience with the particular system we studied.

Subjects classified themselves into decade age ranges, the modal range of which was 25 to 34 years (8 of 19). Most subjects had had some college-level education (13 of 19). The preponderant category of prior work experience was professional secretary-typist (13 of 19); mean reported typing speed of the 19 subjects was 69 words per minute.

Procedure. Subjects were run in groups of 2 to 3 in a simulated office environment. Test rooms were outfitted with wall hangings and phones to appear officelike. The subjects knew they were participating in an experiment, but they also understood that one goal was to realistically simulate an office. Each subject was asked to imagine that a new word processor had just arrived and that they had been assigned the job of working through the training material to learn to use the equipment. We rerouted the phones of several colleagues so that they would ring in the test rooms. Subjects were asked to answer these phones and take messages. They were also given an average of two memos per day to type up on an ad hoc basis. A coffee-break room was established for the subjects, and they were encouraged to use it.

All subjects in a given group were assigned the same manual. At the beginning of the first day, and at the end of each day, an interview and administrative session was held. During the hands-on portion of the experiment, the subjects read and followed the training exercises in their manuals, and periodically (after the completion of specified groups of training topics) were given performance tasks. There was a total of eight performance tasks, described in Figure 4. Tasks 1 through 3 addressed basic topics: document creation, revision, and printing. Tasks 4 through 8 addressed more advanced topics: multipage documents, automatic spell checking, headers and footers, alternate formats, block text moves, and the duplication of system and data diskettes. The material for some of the advanced topics went beyond the scope of the MM and SS. For the advanced topics, subjects were provided with an

***Figure 4.* Description of eight performance tasks in Experiment 1.**

Task 1: Create, paginate and print a one-page letter. Required business-letter formatting using line breaks and punctuation.
Task 2: Create, paginate and print a two-page letter. Similar to Task 1, but also required underlining and simple indentation.
Task 3: Revise, paginate, and print the letter created in Task 2. Required insertion, replacement, and deletion of text, transposition of text, and movement of a paragraph from one page to another.
Task 4: Create, spell check, paginate and print a three-page report. Required centering, a running header and footer, both double spaced and single spaced text, and indented paragraphs.
Task 5: Revise, paginate and print the report created in Task 4. Required insertion, replacement, and deletion of text, revision of the header text, and movement of a paragraph within a page.
Task 6: Revise a four-page document already created and stored on diskette. Required insertion, replacement, and deletion of text, movement of a paragraph within a page, insertion of a regular paragraph, insertion of an indented paragraph, combining two paragraphs into one, addition of a running header and footer, and underlining.
Task 7: Duplicate a diskette containing documents.
Task 8: Duplicate a program diskette.

additional self-instruction training manual; the same advanced manual was provided to both the MM and SS subject groups. Subjects also had access to the entire system library of reference manuals.

If subjects experienced difficulties during learning, they were permitted to telephone a simulated hot-line service. When using this service the subject described a problem and received help. This hot line was staffed by a blind expert, an expert in the system who was very familiar with operational procedures in commercial hot lines for word processing but who was unaware of the goals of the experiment. Additionally, subjects assigned to the same group sometimes worked together on problems they experienced. We do not present an analysis of the use of these learner supports, but we included them in the procedure because both are typical of real on-the-job learning of office information systems, the sort of learning the manuals are designed to support. We had no further procedure for problem support, but if a subject had a problem so severe that nothing seemed to help and considerable time elapsed without progress, we intervened enough to get things moving again.

After finishing the prerequisite portion of the training material, each subject was allowed to undertake the corresponding performance task. During the performance task, the subject could refer to the training material (or the system reference library, or even place a call to the hot line). Learners could work on only one performance task at a time, and could not return to a task once they had pronounced it finished.

Figure 5. **Overall performance measures in Experiment 1.**

	Minimal Manual	Standard Self-Instruction
Learning time (hr), Tasks 1 through 3	5.1	8.5
Learning time (hr), Tasks 4 through 8	4.9	7.9
Performance success (subtasks), Tasks 1 through 8	28.9	10.2
Performance success (subtasks), Tasks 1 through 3	12.1	8.1
Performance time (min), Tasks 1 through 3	63.0	86.4
Performance efficiency (subtasks/ min), Tasks 1 through 3	.23	.11

Scoring. Two dependent measures were collected and analyzed in this experiment: (1) time to complete training and performance tasks and (b) performance on eight word-processing tasks. Time was scored as the time required to read and complete the exercises in the training materials. The training materials covered the functions necessary to complete the eight performance tasks.

Each of the eight performance tasks consisted of several subtask components, specific formatting features and revisions that were scored as correct if the subject used the appropriate word-processing function to achieve the desired effect. The correctness of the formatting features was determined by examining the task documented stored on diskette. The correctness of the revisions was determined by examining both the task documents stored on diskette and a record of the keystrokes produced by each subject as they completed the tasks. The performance score for each task was determined by summing the number features and revisions completed correctly.

4.2. Results

Time. Overall, the MM subjects required 40% less learning them than the SS subjects, $t(17) = 3.06, p < .01$. This result suggests that the Minimal Manual design accomplishes one of its design objectives—to make learning faster. This difference was also obtained—and of about the same magnitude—when we looked only at the learning times for the basic topics (the material prerequisite to Tasks 1 through 3), $t(17) = 2.93, p < .01$. What was perhaps more interesting, however, was that the learning time advantage for the MM group persisted also for the advanced topics (the material prerequisite for Tasks 4 through 8), $t(17) = 2.17, p < .05$. As noted earlier, much of the material to be learned for the advanced topics originated in a common self-instruction manual that both the MM and the SS group studied. Thus the fact that the MM group was significantly faster at reading this material is evidence that the manual also better facilitated further learning. Figure 5 presents the summary time and performance results for Experiment 1.

Task Performance. Overall, the MM subjects accomplished 2.7 times as many performance subtasks as the SS subjects, $t(16) = 3.63$, $p < .01$. This analysis compares the performance score totals by subject (one subject in MM was incompletely scored and was deleted from the performance analysis). Part of the difference resides in the fact that some of the SS subjects ran out of time and hence never even attempted some of the advanced tasks. Accordingly, we also looked at performance success for the basic tasks only. Here also the MM group outperformed their SS counterparts, however, in this case by 50%, $t(16) = 2.11$, $p = .05$ (performance means are presented in Figure 5).

We also combined time and performance into a measure of test performance efficiency: accomplishment in the basic tasks divided by the time to perform the tasks (as noted, not enough subjects attempted the advanced material to include it). The MM subjects were more than twice as efficient as the SS subjects, $t(16) = 2.90$, $p < .01$ (see Figure 5).

4.3. Discussion

Clearly, these are strong and encouraging indications that the Minimal Manual design is a better training design than the standard self-instruction manual. The key question that remains is specifically in what aspects of this design the advantage resides. Our second experiment was smaller in the scale of learning examined and less scrupulously realistic in the task-situation we presented to learners; it was designed to address the question of why the Minimal Manual design is better.

5. EXPERIMENT 2

The second experiment was carried out by Mazur and Carroll at the IBM research center in Yorktown Heights, New York. The principal focus of the study was the contrast between a commercially developed, standard self-instruction manual (SS) and the experimentally developed Minimal Manual (MM).

5.1. Method

Design. The experiment had a 2 × 2 between-subjects design. The two independent variables were Manual, either SS or MM, and Instructions, either to "learn while doing" (LWD) or to "learn by the book" (LBB). The LWD learners were given 5 hr during which they were to perform a series of tasks using the system. The LBB learners were given 3 hr to use the manual in order to learn about the system. They were separately given 2 hr to perform a series of tasks using the system.

Subjects. A total of 32 subjects participated in the experiment, 8 in each of the four manual/instruction conditions. A temporary agency selected and screened participants for two qualities: (a) we wanted people who were experienced with routine office work, typing in particular, and (b) we wanted people with little prior computer experience. All of our participants attested to these requirements. Seven had had some exposure to computers in entering data via "canned" programs loaded by others. Both the median and the modal educational level in our participant sample was high school. Job experiences varied widely. The median and the modal amount of full-time office work experience was 5 years. The preponderant category of prior work experience was secretarial; other major job categories were counselor, switchboard operator, bank teller, accountant, and bookkeeper.

We contolled for age by attempting to balance learners across the four manual/instruction groups among six age ranges: 18 to 25, 26 to 35, 36 to 45, 46 to 55, 55 to 65, and over 66 years. We made this pseudorandom assignment on the basis of appearances only, because we did not want to call subjects' attention to age as a possible performance factor at the start of the experiment. Nevertheless, we were able to get fairly uniform distribution, $\chi^2(15) = 8.43$, $p = .9$. Both the median and the modal age range in our sample was 26 to 35 years.

Procedure. The agenda for experimental sessions involved 2 hr of hands-on experience with the system in the morning, an hour lunch break, and then 3 hr of further hands-on work. The final hour of the day included a system concepts comprehension test (for which there were no significant differences). In our initial instructions we identified the main components of the word-processing system and of its reference library. Subjects were encouraged to rely principally on the training manual (MM or SS), but we wanted to be realistic in making the entire system library available to those who might want to use it. (About half the subjects made some use of the reference manuals.)

During the hands-on portion of the experiment, the observer sat with the subject and made detailed notes about activities and outcomes (what the subject was trying to do, what commands or selections were made, what the effects were, how errors were recovered from, etc.). All observations were time-stamped by reference to a digital timer. Subjects were encouraged to verbalize goals, plans, frustrations, and opinions throughout the session.

***Figure 6.* Description of six performance tasks in Experiment 2.**

Task 1: Create and print a letter. Required simple business-letter formatting via line skips and punctuation.
Task 2: Create and print a letter. Similar to Task 1.
Task 3: Create and print a bulletin. Required centering, underscoring a single word, underscoring a phrase.
Task 4: Revise and print a bulletin. Required text block move, text block insertion, word deletion, use of text locate command, and specification of print quantity.
Task 5: Create and print a bulletin. Required keyword indexing of the document, specification of margins, and specification of an alternate format.
Task 6: Revise and print a bulletin. Required use of menu shortcuts via commands and the definition and use of an editing macro.

However, the ground rules were that the observer could not answer questions or provide hints except in extreme situations. If a subject had failed to make progress in recovering from an error for 20 min, the observer would intervene, returning the system to the state directly preceding the error. If a subject seemed to be distressed, the observer would likewise intervene, placing the system in a familiar state (e.g., the Home menu). Three subjects became frustrated enough to ask to be excused from the experiment and were replaced.

There were a total of six performance tasks given to each learner. The tasks were typing exercises that involved use of the system's word-processing function, and were presented in increasing order of difficulty (see Figure 6). For the LBB learners, Task 1 was presented after 1.5 hr of learning (and a half hour before a hour lunch break). Task 2 was given immediately after lunch. Tasks 3 through 6 were presented after a final 1.5 hr of learning. For the LWD learners, Task 1 was presented at the start of the experiment, and as each successive task was completed the next was given (with the exception that Task 2 was given during the half hour immediately after the lunch break, whether or not this was its place in the sequence).

Learners could work on only one task at a time, and could not return to a task once they had pronounced it to be finished. Only when they had told us that a task was complete, could they go on to the next task. They were permitted to use their manual at all times.

Scoring. We scored a variety of dependent measures. Time and success for the six performance tasks were measured analogously to Experiment 1. Time indicated the time to complete a given performance task; success was the number of subtasks correctly completed. The LWD group contrasted with the LBB group and the subjects of Experiment 1 in that their performance times included the time it took them to learn the material as they completed the task. Scores for the systems concepts comprehension test were merely the number of questions correct.

In addition to these standard performance measures, we wanted to more analytically characterize how subjects learned and performed. We examined the allocation of attention and effort during the experiment. We measured the time it took subjects to get the system started, to load system diskettes, to get to the typing area, and to get printed output. During the first 1.5 hr of performance we classified subjects' activities as "reading the book," "working at the display," or "coordinating attention to both." The grain of this analysis was 30 sec (e.g., a subject had to work at the display for 30 continuous sec in order to be scored as working at the display).

We also tabulated the errors subjects made through the course of the experiment, and we measured the amount of time recovery from these errors consumed. Based on our four pilot subjects and the first block of eight subjects, we developed an error taxonomy for the word-processing system. This included 39 of the most frequent and salient errors, organized into six categories: 3 mechanical errors (trouble loading floppy diskettes or finding the on/off switch); 2 manual errors (skipping text or miscoordinating the manual and the display); 16 menu errors (selecting an exotic item or parameter from a menu, misexecuting the Code–Cancel key combination); 14 typing area errors (confusion of insert and replace mode, use of the space bar to advance the cursor); 3 errors which occurred in both menus and in the typing area; and the residual category of miscellaneous. The scoring judgments were made case by case as the experimental session actually occurred. It is important to stress that the error analysis and performance success on the six tasks were independent measures: A subject could have perfect success on all subtasks but still make many errors, or make few errors and fail to successfully complete the performance tasks (e.g., by progressing too slowly).

The Minimal Manual stressed four general error recovery methods (the use of Cancel, Backspace, Reply, and Power Off). We tabulated subjects' successful use of these methods (i.e., episodes in which the use of a recovery method actually led to error recovery) as well as unsuccessful uses. We assessed use of the MM and SS manuals for reference after training by analyzing episodes in which a subject tried to look back at material that already had been covered.

5.2. Results

Success in learning was assessed by the six performance tasks. Figure 7 summarizes the relevant results. Our analysis of the performance tasks consisted of a two-way analysis of variance (ANOVA) with Manual (MM or SS) and Instructions (LWD or LBB) as between-subject factors and Tasks (1 through 6) as a within-subject factor. For each task we scored the number of subtasks successfully completed. MM learners successfully completed 58%

***Figure** 7.* **Overall performance measures in Experiment 2.**

	Minimal Manual		Standard Self-Instruction	
	By Book	While Doing	By Book	While Doing
Comprehension test score	20.4	23.8	14.3	20.4
Performance success (subtasks), Tasks 1-6	11.4	13.6	4.8	11.0
Performance time (min), Tasks 1-3	71.8	121.7	104.9	130.8
Performance efficiency (subtasks/min), Tasks 1-3	.12	.09	.05	.06

more subtasks than did SS learners, $F(1, 28) = 5.31$, $p < .05$. And LWD learners completed 52% more subtasks than LBB learners, $F(1, 28) = 4.48$, $p < .05$. There was no interaction of the two between-subject factors. (There was a main effect of task—not of any interest in itself, and no interactions of task with either of the two between-subject factors.)

We also applied this ANOVA design to performance efficiency: subtasks completed per unit time for each of Tasks, 1, 2, and 3 (too few learners attempted Tasks 4, 5, and 6 to make including them meaningful). Again, we found an effect of manual: MM learners achieved 93% more per unit of time than did SS learners in Tasks 1 through 3, $F(1, 28) = 6.13$, $p < .05$. There was no interaction between Manual and Instructions (we again ignore the main effect of tasks; there were no interactions).

Analyzing Learning. Our performance results demonstrate the learning efficacy of MM versus SS. This offers some support to the design ideas embodied in MM. We also collected data that bear more directly on the question of how MM affords better learning. MM was designed to allow learners to get started faster: to turn the system on, to load diskettes, to get through the menu control structure to the typing area, to print a first document. We measured elapsed time for each of these achievements. As shown in Figure 8, these measurements do indicate that the MM learners were able to get started faster than the SS learners. The MM learners were almost twice as fast to start the system up, and 20% to 40% faster to load diskettes, to reach the typing area, and to print out their first document. However, these differences were statistically significant only in the case of time to start the system up, $F(1, 28) = 5.92$, $p < .05$. (For the time to start the system measure, there was also a significant interaction of Manual with Instructions, $F(1, 28) = 5.52$, $p < .05$, reflecting the fact that the MM/LBB group was particularly fast and the SS/LWD group particularly slow.)

MM was also designed to encourage learners to coordinate their attention between the manual itself and the consequences of their actions on the

Figure 8. **Getting-started benchmarks in Experiment 2.**

	Minimal Manual		Standard Self-Instruction	
	By Book	While Doing	By Book	While Doing
Time to start system (min)	4.9	8.7	16.5	8.9
Time to load diskettes (min)	7.6	8.6	9.9	12.8
Time to reach typing area (min)	40.7	35.4	60.6	33.3
Time to print out a document (min)	42.7	56.3	72.6	53.8

Figure 9. **Allocation of attention in the first 1.5 hr in Experiment 2.**

	Minimal Manual		Standard Self-Instruction	
	By Book	While Doing	By Book	While Doing
Reading the display (min)	27.9	28.5	34.4	38.2
Coordinating manual and display (min)	46.2	38.5	31.1	39.4

system's display. We assessed this by classifying learner activities during the first 90 min of the experiment as "reading the book," "working at the display," or "coordinating attention to both." As Figure 9 shows, there is apparently a trade-off between the first and third of these categories. MM learners tended to spend relatively less time (29% less) reading the manual, but relatively more time coordinating their attention between the manual and the display (20% more), though this difference was nonsignificant.

Yet another design objective of MM was to support detection of and recovery from errors. Figure 10 summarizes our principal measures. MM learners made 20% fewer errors, $F(1, 28) = 3.11$, n.s, and spent 10% less time recovering from errors. The experimenter was forced to intervene less than half as often for MM learners as for SS learners. MM learners successfully used recommended error recovery methods 60% more often than did SS learners. Although all of the mean differences here accord with our predictions, none reached statistical significance.

MM learners did make greater use of the four specific error recovery methods that were stressed in MM, namely, Cancel, Backspace, Reply, and Power Off, $F(1, 28) = 7.48$, $p < .01$. There was a main effect of Recovery Method (the different recovery methods were used with varying frequencies), and an interaction of Recovery Method with Manual, $F(3, 84) = 3.85$, $p < .01$—probably due to assorted asymmetries (Cancel was used disproportionately by the MM/LBB group, Backspace by the SS/LBB group, etc.).

Finally, MM was designed to support reference use after training. During

***Figure 10.* Errors and Recovery in Experiment 2.**

	Minimal Manual		Standard Self-Instruction	
	By Book	While Doing	By Book	While Doing
Errors and their consequence:				
Overall error frequency	187.3	188.8	224.1	261.0
Error recovery time (min)	121.7	153.3	163.7	142.3
Intervention frequency	1.0	.3	2.3	.9
Frequency of different recovery methods:				
Cancel	30.4	19.5	7.0	16.6
Backspace	12.0	12.1	9.1	10.6
Reply	1.0	1.4	1.4	3.8
Power off	3.9	5.0	2.6	1.4

***Figure 11.* References to previously encountered information in manual in Experiment 2.**

	Minimal Manual		Standard Self-Instruction	
	By Book	While Doing	By Book	While Doing
Total references per subject	20.6	35.8	20.9	35
Successful references	10	13	5.6	11.6

the experiment, we noted every reference made to previously encountered information in the training manuals (i.e., reference to a section of the manual after it had been studied). We classified each reference as successful or unsuccessful according to whether the goal that prompted the reference to the manual was in fact satisfied by the reference. For each learner we then computed the ratio of successful references to the manual to the total number of references to the manual. This ratio was reliably larger for the MM learners, $F(1, 28) = 4.42$, $p < .05$ (see Figure 11).

A second indication of this difference was the number of subjects who elected to make use of the system's reference library in the course of the experiment. None of the tasks we gave them required them to use any manual other than the training manual (MM or SS), but 17 of the 32 did so; 13 of these 17 were SS subjects, $p < .005$ (by the Fisher Exact Test). SS subjects spent over 20 min on average referring to reference manuals in the library, whereas MM subjects spent an average of 2 min.

Targeted Errors and Skills. We felt that our empirically developed error taxonomy worked well for the analysis of error in this study. Only 250 (less than 4%) of the errors were classified miscellaneous. The distribution of the 6,885 observed errors among our 40 error types was very skewed (see Carroll

Figure 12. **Frequencies per subject of major errors in Experiment 2.**

	Minimal Manual		Standard Self-Instruction	
	By Book	While Doing	By Book	While Doing
Exotic choice	13.5	16.6	26.5	19.6
Exotic parameter	14.8	19.4	27.9	22.5
Code-Cancel	9	16	26.3	22.4
Enter to exit a menu	20.5	16.9	21.4	22.3
Keystroke bursts	10.8	9.4	21	44

et al., 1986, for details). Five error types seemed particularly important—alone they accounted for over 46% of the errors; all were at least 50% more frequent than the sixth most frequent error. The five errors were Exotic choice (selecting irrelevant and advanced menu branches), Exotic parameter (altering menu defaults needlessly), Code-Cancel (miscoordinating the compound keypress for Cancel), Enter to exit (trying to leave a menu without having made any selection), and Keystroke bursts (seemingly random keystrokes, usually to recover from errors).

The first three of these were errors that the MM design specifically targeted. They were important errors: Learners spent an average of 36 min recovering from the direct consequences of making these three errors, or 25% of the average total amount of error recovery time. (On average, subjects spent 145 min—almost half their time—recovering from errors.) We performed a separate ANOVA for the sum of the frequencies of these three errors with the between-subjects variables of manual (SS or MM) and condition (LWD or LBB). There was a significant effect of manual, $F(1, 28) = 5.46$, $p < .05$, indicating that the MM group committed these errors less often (see Figure 12).

Other specific errors targeted in the MM design were statistically more minor. Skipping text was 40% less frequent for the MM learners; miscoordinating the manual and screen was 13% less frequent. However, together these two errors constituted less than 3% of the total errors.

As described earlier, most of the MM topic chapters had On Your Own open-ended exercises. Each of the MM learners spent an average of 14.2 min on these exercises (this did not differ greatly between LBB and LWD). The majority of the learners tried the On Your Own exercises for the first four topics, but very few of them tried any of the later exercises.

Subjective Measures. The final portion of the system concepts comprehension test consisted of attitude questions that allowed subjects to express their feelings about this learning experience. For example, we asked them if there were things they would have liked to learn but did not get to cover, and whether they would have approached the task differently if they had been learning to use the system in some other situation. A subset of the questions, however, was directed more specifically at their attitudes about overall learning difficulty in the experiment. We asked them to imagine a 10-week course in office skills, and to allot time for learning to use the word-processing system. Both the median and modal response for the SS subjects was 200 hr (or 50% of the 10-week course), for MM both the median and the mode were 80 hr (or 20%), $p < .05$ (by Mann-Whitney U-test, $U = 66.5$ – only 29 learners answered). In both Manual groups, the LBB learners estimated briefer times than did the LWD learners (LBB in SS estimated 120 hr, whereas LWD estimated 200 hr; LBB in MM estimated 80 hr, whereas LWD estimated 165 hr).

We asked the learners whether they had expected that learning to use a word processor would be difficult. Twenty-two learners answered "yes," 8 answered "no." There were no differences between the manual or instruction groups on these reported initial expectations. We asked them whether, in view of their experience in our study, they now thought that learning to use a word processor was more or less difficult than expected. The 13 SS learners who made a signed judgment were split on this question (7 more difficult, 6 less difficult), but the 13 MM learners who made a signed judgment overwhelmingly judged it to be less difficult (12 to 1), $p < .05$ (by Fisher Exact Test).

5.3. Discussion

In this experiment, MM subjects performed better and more efficiently than their SS counterparts. More importantly from an analytical perspective, they did the things the Minimal Manual was designed to facilitate: They got started faster; they coordinated attention better; they made fewer errors; in particular, errors the Minimal Manual targeted and trained against; they made better use of error recovery methods; they made better use of the training manual for later reference. The present experiment shows not only that the manual works well, it begins to show in detail why it works well.

Many of the manual differences we predicted emerged more clearly between the LBB groups than they did between the LWD groups. For example, looking only at LBB subjects, the MM group was 1.4 times faster than the SS group to start the system, 1.5 times faster to get to the typing area, and 1.7 times faster to print out a document (see Figure 8). During the first 1.5 hr of learning, the MM group coordinated attention between the manual and the screen 49% more than did the SS group (see Figure 9). Finally, under LBB instructions the MM group spent 35% less time in error recovery than the SS group (see Figure 10). Nevertheless, the interactions of manual and instructions were generally marginal or nonsignificant due to the great intersubject variability.

In an earlier study, Carroll and Carrithers (1984) suggested a navigational analogy to understand the contrast between LWD and LBB:

> Learning by the book is like navigating unfamiliar territory by following a list of very specific treasure-map instructions ("March ten paces east, then turn toward the large oak tree"). The task is well defined until you miss a checkpoint, and then you don't know where you are. Learning by doing is like navigating unfamiliar territory by the stars. Many errors will be committed (because following a star can lead you to cliffs and rivers), but correction is always possible by simply looking up at the sky. (p. 386)

Some results of the present study are consistent with this view: LWD had a 9% higher error rate overall than LBB, but spent a less than proportional amount of time in error recovery (3% more than LBB) and, perhaps, more importantly, required only a third as many experimenter interventions as LBB (see Figure 10). Unfortunately, these are nonsignificant trends in the data, and hence our experiment fails to further clarify the difference between learning while doing and learning by the book.

6. GENERAL DISCUSSION

In our introduction, we described the troubled state of the art in user self-instruction, particularly in training manuals. A variety of approaches are being explored, such as including advanced organizers (Foss, Rosson, & Smith, 1982) and presenting diagrammatic frameworks for training exercises (Galambos, Sebrechts, Wikler, & Black, 1984). In our own prior work, we have experimented with a guided exploration situation (Carroll et al., 1985) in which learners were given no manual at all but rather a set of unordered cards, each of which was directed at a particular user-pertinent goal (e.g., typing something, quitting work), but which provided only hints about how to accomplish the goal and error recognition and recovery information for when the goal was not accomplished. As mentioned earlier, our guided exploration learners sometimes explicitly voiced a desire to have a real manual. The Minimal Manual attempts to capitalize on the strengths of guided exploration and the desire of learners to have a more traditionally structured manual.

We wanted to design a training manual in the self-instruction genre, but one that allowed users to get started doing recognizably real work, one that deemphasized reading in favor of action, and one that helped learners to avoid making errors and to recognize and recover from errors committed. Our two experiments converge on the conclusion that the Minimal Manual is substantially and reliable superior to the commercial self-instruction manual. The bases for this advantage, insofar as we can assess them now, accord with our specific design objectives. This suggests a simple and rather direct path from our current understanding of the learning problems of new users to the Minimalist training model that can address these problems. Facilitating the tasks that learners already understand and are motivated to work on, slashing the instructional verbiage they must passively read, and addressing important user errors can produce better training material than the current state of the art.

Interestingly, the Minimalist slogan "Less can be more" also may extend to the time and cost required to develop training. The analytic and subskill phases of design for the Minimal Manual together required less than a man-month of effort (Carroll, 1984). This was a retrofitted design (MM was built out of SS), but this retrofitting created both savings and idiosyncratic obstacles: It is sometimes easier to modify an actual exemplar than to create anew, but the particular demand we placed on ourselves to have MM and SS differ only in the ways described in the introduction definitely cost time and effort that would not have been required if we had sought only to design a good training manual. In any case, the practice of developing one training manual out of another is typical in the computing industry (the manual for Level 2.3 of a software product is developed from the manual for Level 2.2, the manual for the new word-processing system is developed from that for the old one, etc.).

Many questions remain for further research and analysis, among these the questions of how far the present results can be generalized and of what specific cognitive processes underlie the learning differences between the Minimal Manual and the self-instruction manual. It may be quite important that we studied office workers learning work-related procedural skills. Our conclusion may not generalize to other areas of educational technology. It may even be significant that we studied clerical workers learning word processing. To assess the generality of the advantage of Minimalist instruction, we need to expand the investigation to other instructional domains. This is happening. Ongoing work by Olfman (1987) at Indiana University has, for example, developed a Minimalist manual for a spreadsheet application.

Perhaps the greatest opportunity for examining boundary conclusions for the utility Minimalist instruction lies in helping development teams use this model in their work. Developers have to use some instructional model, and the standard self-instruction model is associated with a variety of learner problems, as described in this article. Thus the interests of research and development can converge on the strategy of regarding current development work as "appropriately scaled" research. Some of our current effort is directed at helping such development experiments take place.

Although the present studies demonstrate learning advantages of the Minimalist approach, we need to know more about the specific cognitive bases of these advantages to understand more precisely how the Minimalist approach works. Currently, we are directing some of our effort at contrasting alternate minimal manuals, each of which incorporates some but not all of the design ideas of the Minimal Manual studied here. For example, in the design of the Minimal Manual we deliberately specified some procedures incompletely, to involve the learner more in understanding and executing the procedures, and to encourage the coordination of attention between the system and the training. Black, Carroll, and McGuigan (1987), using multiple alternate minimal manuals, showed a more specific effect of incomplete instructions, namely, the benefit of forcing learners to make inferences (vs. forcing covert rehearsal of procedures). We are now investigating the specific effects of different types of inferences on learning.

More broadly, we need to understand how standard self-instruction models, based on educational prescription and experience, produced training materials that were substantially inferior to our Minimalist materials. We believe that there are specific properties of the standard models that caused this difference. The "systems approach" of Gagne and Briggs (1979; see also Mager, 1975) is focused on producing a comprehensive fine-grained analysis of instructional objectives, a reasonable starting point for instructional design. However, it does not provide guidance in other critical areas. For example, it provides no guidance for controlling instructional verbiage; indeed, the emphasis on comprehensive decomposition of instructional objectives naturally leads to "maximalist" content. The systems approach prescribes curricula of accretional lesson sequences for the presentation of training material, not recognizably real-task scenarios for learning. Indeed, learners are asked to periodically "demonstrate criteria," that is, to perform rote exercises in order to advance through the training curriculum (Mager, 1975). Learner motivation and attitude are merely "assumed to be present" (Gagne & Briggs, 1979) but never addressed directly by the instructional design itself. Relevant prior knowledge is similarly ignored. Carroll and Herder (in preparation) analyze in more detail how the standard model differs from Minimalist design and the consequences of these differences.

We believe that this design project has succeeded to this point because it originated both in a detailed qualitative analysis of the problems real users of word-processing equipment have in realistic learning situations and in a theoretically grounded analysis of possible intervention strategies (the Minimalist training model). We suggest that when applications of cognitive science fail, it is frequently because they have addressed problems that only seem plausible in laboratory tasks. In this regard, it is important to bear in mind that the problems we identified in our design analysis of learning problems in the word-processing domain are by and large not the problems that the designers of self-instruction training manuals set out to address.

Acknowledgments. We thank Georgia Gibson for helping to organize and run Experiment 1. We are grateful to Judy Olson, Tim O'Shea, Don Norman, Mary Beth Rosson, and John Thomas for comments.

REFERENCES

Black, J. B., Carroll, J. M., & McGuigan, S. M. (1987). What kind of minimal instruction is the best? In J. M. Carroll & P. P. Tanner (Eds.), *Proceedings of CHI+GH'87: Conference on Human Factors in Computing and Graphics Interface,* 159–162. New York: ACM.

Carroll, J. M. (1984). Minimalist training. *Datamation, 30/18,* 125–136.

Carroll, J. M. (1985). Minimalist design for active users. In B. Shackle (Ed.), *Human-Computer Interaction—INTERACT '84,* 39–44. Amsterdam: North-Holland.

Carroll, J. M., & Carrithers, C. (1984). Blocking learning error states in a training wheels system. *Human Factors, 26,* 377–389.

Carroll, J. M., & Herder, R. E. (in preparation). Why didn't Instructional Science help?

Carroll, J. M., Mack, R. L., Lewis, C. H., Grischkowsky, N. L., & Robertson, S. R. (1985). Exploring exploring a word processor. *Human Computer Interaction, 1,* 283–307.

Carroll, J. M., & Mazur, S. A. (1986). LisaLearning. *IEEE Computer, 91*(11), 35–49.

Carroll, J. M., & Rosson, M. B. (1985). Usability specifications as a tool in iterative development. In H. R. Hartson (Ed.), *Advances in human-computer interaction* (pp. 1–28). Norwood, NJ: Ablex.

Carroll, J. M., Smith-Kerker, P. L., Ford, J. R., & Mazur, S. A. (1986). The Minimal Manual. *IBM Research Report* (Research Rep. No. 11637). Yorktown Heights, NY: IBM.

Davis, J. (1984). What are the users doing? *Seybold Report on Office Systems, 7,* 11–18.

Dreyfuss, H. (1955). *Designing for people.* New York: Simon & Schuster.

Eason, K. D. (1976). Understanding the naive computer user. *The Computer Journal, 19,* 3–7.

Foss, D. A., Rosson, M. B., & Smith, P. L. (1982). Reducing manual labor: Experimental analysis of learning aids for a text-editor. *Proceedings of Conference on Human Factors of Computer Systems,* 332–336. Gaithersburg, MD: National Bureau of Standards.

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

Galambos, J. A., Sebrechts, M. M., Wikler, E., & Black, J. B. (1984). A diagrammatic language for instruction of a menu-based word processing system. In S. Williams (Ed.), *Humans and machines: The interface through language* (pp. 11–44). Norwood, NJ: Ablex.

Mack, R. L., Lewis, C. H., & Carroll, J. M. (1983). Learning to use word processors: Problems and prospects. *ACM Transactions on Office Systems, 1,* 254–271.

Mager, R. F. (1975). *Preparing instructional objectives.* Belmont, CA: Pitman Learning.

Olfman, L. (1987). *A comparison of construct-based and applications-based training methods for DSS generator software.* Dissertation proposal, Indiana University, Bloomington.

Piaget, J. (1985). *The equilibration of cognitive structures: The central problem of intellectual development.* Chicago: University of Chicago Press.

Reder, L. M., & Anderson, J. R. (1980). A comparison of texts and their summaries: Memorial consequences. *Journal of Verbal Learning and Verbal Behavior, 19,* 121–134.

Schank, R. C., Collins, G. C., & Hunter, L. E. (1986). Transcending inductive category formation in learning. *Behavioral and Brain Sciences, 9,* 639–686.

Scharer, L. L. (1983). User training: Less is more. *Datamation, 29/7,* 175–182.

Uhler, H. L. (1981). Training and support: Shifting the responsibility. *Seybold Report on Word Processing, 4,* 1–10.

Uhler, H. L. (1984). Training—managers, professionals, executives. *Seybold Report on Office Systems, 7,* 1–10.

HCI Editorial Record. First manuscript received August 7, 1986. Revision received July 10, 1987. Accepted by Tim O'Shea—*Editor*