

Designing Interactive Tutorials for Computer Users

Davida H. Charney
The Pennsylvania State University

Lynne M. Reder
Carnegie-Mellon University

ABSTRACT

The aim of this article is to find the optimal combination of written instruction and on-line practice for learning a new computer application. Subjects in the experiment learned commands for an electronic spreadsheet by reading brief user-manual descriptions and working training problems on-line. The form of the training problems was varied in a within-subjects design to control how much independent problem solving subjects engaged in while learning any given command. There were three forms of practice: (a) pure guided practice, in which subjects were told exactly what keystrokes to type to solve the problems; (b) pure problem-solving practice, in which subjects solved problems without guidance; and (c) mixed practice, in which the first problem for a command was presented in guided practice form and two others in problem-solving form. The spacing of the training problems was also manipulated; the problems pertaining to a given command were either massed (i.e., presented consecutively) or distributed (i.e., separated by other instructional material). After a 2-day delay, subjects solved new problems on the computer without referring to the instructional materials. The results indicate that problem solving was a more difficult form of training than guided practice, but it produced the best performance at test. Distributing the spacing of training problems during train-

Authors' present addresses: Davida H. Charney, Department of English, The Pennsylvania State University, University Park, PA 16802; Lynne M. Reder, Department of Psychology, Carnegie-Mellon University, Schenley Park, Pittsburgh, PA 15213.

CONTENTS

1. STRATEGIES FOR SKILL LEARNING
 2. EXPERIMENTAL METHOD
 - 2.1. Design
 - 2.2. Materials
 - 2.3. Subjects
 - 2.4. Procedure
 3. RESULTS AND DISCUSSION
 - 3.1. Scoring
 - 3.2. Performance on Problems During Training
 - 3.3. Performance on Problems at Test
 - 3.4. The Control Group Versus the Experimental Group
 - 3.5. The Disadvantages of Guided Practice Training
 - 3.6. Problem Solving Versus Mixed Practice
 4. CONCLUSION
 - 4.1. Discovery Learning and Learning by Example Revisited
 - 4.2. Speculations and Implications
-

ing also improved performance at test. The results have clear pragmatic implications for the design of interactive tutorial manuals as well as implications for cognitive models of skill acquisition.

1. STRATEGIES FOR SKILL LEARNING

Designers of instructional manuals for computer users are beginning to recognize the importance of cognitive models of text comprehension and skill learning for making decisions about such issues as information content, organization, layout, and so on. This trend is all to the good; however, problems arise when different learning paradigms lead to conflicting principles for text design. For example, two powerful learning strategies that have received great attention in recent years are *learning from examples* and *discovery learning*. These strategies suggest quite different approaches to the content of instructional texts. The learning from examples approach suggests that the text provide numerous worked-out examples that learners can use as models for solving problems on their own. Sweller and Cooper (1985) advocated this approach for teaching math students to apply algebra procedures appropriately. In contrast, Carroll, Mack, Lewis, Grischkowsky, and Robertson (1985) argued that people learning to use a word processor should set their own goals and explore the workings of the system with minimal guidance from the text. Carroll and his colleagues even suggested omitting information about some features of the system from the manual, in order to force learners to discover the information independently.

Carroll et al.'s (1985) discovery learning approach is a direct response to two aspects of standard commercial manual design. The first aspect concerns the degree of explicit explanation. In the traditional view, instructional manuals for novice users should be as complete and explicit as possible, containing detailed explanations of every relevant concept and procedure (Price, 1984; Tausworthe, 1979). However, as many people have observed, computer users generally dislike reading long, detailed manuals; they prefer to have someone show them what to do or figure things out for themselves (Carroll, 1984; Draper, 1984; Scharer, 1983). Carroll (1984) found that learners actually performed better on on-line tests after studying drastically shortened manuals that eliminated large sections of explanation and elaborations than after studying a full-length commercial manual. In contrast, Reder, Charney, and Morgan (1986) prepared two versions of a manual for the PC-DOS operating system, one version with elaborations and the other without. Users who studied the elaborated version performed better on an on-line test than did those who studied the unelaborated manual. However, when the availability of elaborations was controlled by type, experienced and inexperienced computer users were found to benefit only from selected types of elaborations. We will return to these results later.

A second common feature of commercial manuals is the so-called "tutorial" section that provides users with directed, step-by-step, hands-on practice. In many respects, tutorials are quite similar to the presentation of worked-out examples; the major difference is the addition of the motor activity of carrying out the step-by-step instructions and viewing the system's prompts and feedback. Carroll et al. (1985) argued that people learn skills best through exploration and that instead of describing each system feature explicitly and guiding learners through tutorial exercises, manuals should leave many details of the system to be discovered by the learner. In a study of people learning to use a text editor, Carroll et al. demonstrated the superiority of a "guided exploration" manual over a tutorial manual. The guided exploration materials described each procedure very briefly, with minimal explanation, and even omitted certain procedural details. Subjects using the guided exploration materials set their own "problems" (e.g., deciding to compose and print a letter) and executed procedures at their own initiative. In contrast to people working through a commercial tutorial manual, subjects who worked with the guided exploration materials spent less time on training, less time on the test (typing and printing a letter), and used the procedures more successfully (fewer failed attempts at executing a command).

We share the intuition that people who follow instructions step-by-step often work mechanically, without thinking enough about the purpose of each action. However, it is not clear that discovery learning is the best alternative to step-by-step instruction or indeed that it was the primary cause of the improvement that Carroll et al. (1985) found. As the authors themselves pointed out, there were

substantial differences in the information content, clarity, and organization of the two manuals, so it is difficult to know how much the discovery aspect of the experimental materials contributed to the results. Second, discovery learning was confounded in this study with problem-solving activity. The subjects in the exploration group not only set their own goals, but they also worked independently to solve them. Because problem solving is itself a powerful learning tool, it may have been this rather than the goal setting and exploration that contributed most to learning.

There are several reasons why discovery learning may not be the optimal way to learn a new system. An important part of learning a new system is knowing when any given procedure is most appropriate (Charney & Reder, in press; Reder et al., 1986). This kind of knowledge is often difficult to acquire without previous exposure to the problem situations that may commonly arise. Users learning an operating system or a programming language might have difficulty setting reasonable goals that really explore the system's capabilities. For example, a user who views a text editor as a glorified typewriter may not be able to invent a problem involving multiple windows or search-and-replace functions. Further, inexperienced users may never invent a problem for themselves that demonstrates the advantages of one procedure over another. As we argued elsewhere (Charney & Reder, in press), unless learners are presented with the situations that motivate the use of one procedure over another, they are likely to stick with some procedure they find most memorable, regardless of its efficiency. Finally, novice users who learn primarily through exploring a system may develop and retain serious misconceptions unless their exploration leads to a highly salient error or problematic result.¹

Sweller and Cooper (1985) had similar doubts about problem solving itself, which is an integral part of discovery learning. They argued that learners benefit more from studying worked-out examples than from problem solving, at least early in the learning process. Problem solving, they contended, interferes with the acquisition of schemas of problem types which are necessary for learning to use procedures appropriately. For example, an expert at solving algebra problems has schemas for discriminating between problem types and knows equations that are suitable for each type. Unless novices acquire similar schemas, they must fall back on standard problem-solving search techniques to find appropriate procedures to use. They may spend a long time on fruitless solution paths without ever discovering the "right" one and may never extract a general rule from their experience. Studying worked-out examples of math problems gives students the necessary information for discriminating types of problems and allows them to build appropriate schemata. Consistent with this analysis, Sweller and Cooper found that math students who studied four exam-

¹ As discussed, for instance, in a talk by Christine Neuwirth, "Toward an Intelligent Tutoring System for Effective Written Sentences," presented at the Conference on College Composition and Communication, Minneapolis, March 1985.

ple problems and solved four problems during training were faster and more accurate at solving new problems at test than students who trained by solving eight problems.

The apparent conflict in the results from the discovery learning and learning by example paradigms may be due to a difference in what aspect of skill learning is under investigation. Studying examples may be important for certain components of skill learning, whereas discovery learning may be important for others. To explore this possibility, we must consider more carefully the types of learning that go on in initial skill acquisition. We conceive of initial skill learning as consisting of three critical components (Charney & Reder, in press; Reder et al., 1986):

- learning novel concepts and the functionality of novel procedures;
- learning how to execute the procedures;
- learning the conditions under which a procedure is applied, and remembering the best procedure to execute in a given situation.

In light of this analysis, the results of Carroll et al. (1985) seem to concern primarily the first two components: discovering what options/features are available on a system and how to implement them. Perhaps because the execution of mathematical operations is a well-learned skill for high school students, Sweller and Cooper's (1985) results seem more concerned with the third component, recognizing which procedures are appropriate in particular situations. In our earlier research (Reder et al., 1986), we found that learners benefited from elaborations and examples concerning the second component, namely, how to execute procedures, but not from elaborations on the concepts or when to apply procedures. Our results suggested that manuals with examples of correct commands are superior to manuals without such elaborations, but we had not contrasted reading examples against carrying out tutorial exercises or against problem solving (with or without a discovery component).

The present experiment contrasted various combinations of written instruction and active problem solving during training in learning a cognitive skill, namely, learning to use an electronic spreadsheet. To learn some spreadsheet commands, subjects worked through solutions to training problems step-by-step in a tutorial format (pure guided practice). For other commands, they solved training problems independently (pure problem solving). For a final group of commands, subjects received a combination of guided practice and problem solving (mixed practice). Finally, a group of control subjects studied the complete set of training problems, all with explicit solutions, but did not type anything at the computer.

We expected that guided practice would be of greater benefit to novice computer users than simply studying example problems and their solutions (control group). Novices often have difficulty understanding the purpose of a procedure

or the significance of different parameter specifications until they have seen the procedure demonstrated. Studying the steps of an example problem without actually carrying them out may not be sufficient for comprehension and long-term retention. As noted earlier, carrying out step-by-step instructions also provides information about the system's prompts and feedback. We expected guided practice to exhibit some of the benefits of examples, such as helping users learn to distinguish between procedures. On the other hand, we expected problem-solving practice to help users learn to generate correct sequences of operations, which they may not learn well unless they are faced with problems to solve without guidance. Because guidance seems important for some aspects of skill learning, and independent problem solving seems important for others, the best performance might arise in the mixed practice condition, which provides an initial problem in tutorial form followed by opportunities to solve novel problems independently.

Before describing the study in detail, we would like to highlight briefly the ways in which it differs from Sweller and Cooper (1985) and Carroll et al. (1985). The design of our study differs from that of Sweller and Cooper in several important respects. First, Sweller and Cooper did not have a condition analogous to our guided practice condition. Their subjects studied sample algebra problems but did not work through them with step-by-step guidance. Second, Sweller and Cooper did not have a pure example training condition. In order to motivate their subjects to attend to the examples, they paired each worked-out example with a similar problem for the subjects to solve independently. Finally, Sweller and Cooper's subjects were tested immediately after the training session, whereas we imposed a 2-day delay between training and testing. Our study makes a more rigorous comparison of learning by examples and problem solving because it includes both pure and mixed training conditions. We imposed a delay between training and test because we suspected that the benefit of problem solving may increase with time, as retrieval of particular examples from memory becomes more difficult.

Like Carroll et al. (1985), our study contrasts tutorial practice against problem solving for learning a computer application. However, our design differs from theirs in two major ways. First, we held the information content of the manuals constant across conditions and only varied the type of practice users engaged in. Second, our subjects were provided with opportunities for problem solving but not for discovery learning. Our claim is that a substantial part of the advantage that Carroll et al. attributed to discovery learning may be due simply to the problem-solving activity their subjects engaged in. We do not test this claim directly because we do not have a discovery learning training condition. However, our claim will be indirectly supported to the extent that problem solving without goal setting and exploration produces better performance than does pure guided practice.

Our study also had a secondary goal that neither of the other studies shared;

namely, to extend to the cognitive skill domain the well-known result of superior learning with distributed practice. In numerous verbal learning experiments, subjects have been found to perform better on recall tests if the training list distributed the repetitions of the items across the list than if the repetitions appeared in the list consecutively (for reviews, see Glenberg, 1979; Hintzman, 1976). We orthogonally varied the spacing of practice and the type of practice to see whether increased problem-solving activity attenuates the benefit of distributed practice.

2. EXPERIMENTAL METHOD

2.1. Design

This study involved two groups of subjects, one considered experimental and the other control. The experimental subjects learned commands for the VisiCalc electronic spreadsheet by reading manual entries and working practice problems on the computer. The control subjects read the same text as the experimental group and studied the same training problems (all with explicit solutions), but were not permitted to type anything at the keyboard during the training session.

For all subjects, the text describing the commands was held constant, but the nature of the training problems and their spacing varied orthogonally. The Spacing factor determined whether the training problems for a given command were presented consecutively (massed), or whether text and problems concerning other commands intervened (distributed). The spacing variable was a within-subject variable, but the practice form variable was partly between and partly within subject. The control subjects saw only one form of training problem, that is, with an explicit worked-out solution. The experimental subjects saw different forms of training problems for different commands. There were three forms of practice: (a) pure guided practice, in which subjects were told exactly what keystrokes to type to solve three training problems; (b) pure problem-solving practice, in which subjects solved the training problems without guidance; and (c) mixed practice, in which the first training problem for a command was in guided practice form and the remaining two problems were in problem-solving form.

The last factor, Command Difficulty, was within-subject, varying for both the experimental and control groups. The commands were classified into two groups, "difficult" and "easy." Commands of both difficulty levels were randomly assigned to practice form and spacing conditions for each subject.

For both experimental and control groups, learning was measured in a delayed test in which subjects solved additional problems on the computer without reference to the training materials. The main dependent measures were success at solving the problems and time at task.

2.2. Materials

Training Manuals. Individual training manuals were prepared for each subject. Each manual began with a general introduction to the VisiCalc electronic spreadsheet (725 words), including instruction in scrolling the spreadsheet and moving the cursor. Next came brief descriptions (averaging 250 words) of 12 VisiCalc commands (e.g., entering and formatting data, deleting rows, replicating entries). Six commands were classified as difficult because their syntax was complex, and six, with relatively simple syntax, were classified as easy. These are listed in Figure 1. The 12 commands were independent in the sense that learning the syntax for one would not facilitate acquisition of another. The descriptions contained explicit information about the purpose of each command and an abstract rule for the command syntax. The parts of the rule were explained in detail, but no examples of correct commands were provided. A typical manual entry for an easy command (blank) and a difficult command (move) appear in Figures 2 and 3, respectively. The manual entries for each command were followed by one or more training problems, presented individually on separate pages.

Training Problems. Four problems were constructed for each command; one was randomly selected to be the final test item for a given subject and the remaining three appeared in the manual as training problems. A typical problem presented the user with a previously prepared VisiCalc spreadsheet on the screen of the computer and a goal for how the spreadsheet should be modified. The solution to the problem required just one of the 12 target commands, but also may have required standard scrolling and cursor movement operations.

The experimental design required two versions of the training problems: a version that presented a step-by-step solution (guided practice form) and a version that simply presented a goal that subjects were to achieve on their own (problem-solving form). Recall that the mixed practice condition did not present a new kind of problem; in this condition, subjects learned about a command by working one guided practice problem followed by two problem-solving problems. Figure 4 presents the two versions of the instructions for a problem pertaining to the move command and the associated VisiCalc display.

As indicated in Part A of Figure 4, the guided practice form of a problem stated the goal without naming a specific command for achieving it. The goal was followed by the instruction "TYPE THIS" and a solution to the problem, presented as a sequence of keys to press. In problem-solving form (Part B of Figure 4), the wording of the goal was preserved as much as possible, and the instructions for what to type were simply omitted. All training problems that required problem solving provided feedback on the next page. In order to keep constant the amount and kinds of presented information across conditions, the feedback consisted of the sequence of keystrokes provided in the guided practice

Figure 1. VisiCalc commands presented in the training manual, classified by difficulty.

Easy Commands		Difficult Commands	
/B	Blank a cell	(type)	Enter value or label in a cell
/D	Delete a row or column	/E	Edit the entry in a cell
/F	Format a cell's entry	/M	Move a column or row
/GC	Change width of columns	/PF	Create a printfile on disk
/T	Transfix column/row headers	/R	Replicate a cell or cells
/-	Fill a cell with characters	/W	Split window

Figure 2. A typical manual entry for an easy command (Blank).

Blank Command	
The Blank command irretrievably erases the entry in the current cell.	
<i>Procedures:</i>	
/B [RETURN]	Erases the label or value in the current cell.
If you typed /B by mistake, you can abort the command as long as you have not yet typed [RETURN]. To undo the /B, type any key <i>except</i> [RETURN], [HOME] or an arrow key.	

version of the problem. To avoid mentioning the names of the commands directly, the instructions occasionally included a diagram of the goal-state of the spreadsheet. In the problem-solving form of these problems, the diagrams appeared on the same page as the instructions, but in guided practice form, they appeared as feedback on the following page. Subjects in all conditions saw the same VisiCalc spreadsheet on the screen when working on a problem (as in Part C of Figure 4).

The version of a training problem that a subject saw varied with condition. Subjects assigned to the control group always saw training problems in guided practice form, but they did not carry out the solutions; they merely studied them. For each subject in the experimental group, commands were randomly assigned to practice forms, with the constraint that four commands were assigned to pure guided practice, four to pure problem-solving practice, and four to mixed practice. The assignment was further constrained to ensure that two of the commands assigned to each form were difficult and two were easy.

As described in Section 2.1., spacing of problems was a within-subject variable. For commands assigned to the massed condition, the three training problems appeared immediately after the relevant manual entry. For commands in the distributed condition, one problem appeared immediately after the relevant

Figure 3. A typical manual entry for a difficult command (Move).

Move Column or Row

The Move command moves the entire row or column that contains the current cell to another position on the worksheet.

Procedures:

<code>/M [FROM] . [TO] [RETURN]</code>	Moves the contents of row or column in the [FROM] coordinate to the row or column specified in the [TO] coordinate.
--	---

The Move command requires the following information:

- **The FROM Coordinates:** The coordinates of a cell in the row or column that you wish to move. VisiCalc automatically fills in the coordinates of the current cell (e.g., D5) as the FROM coordinates. If the current cell is not in the row or column you wish to move, type [BKSP] to erase these coordinates and type the coordinates of a cell in the row or column you want to move. Then type a period. Three periods appear on the edit line. Now you can type the "TO" coordinates.
- **The TO Coordinates:** The coordinates of a cell specifying the destination of the move. The TO coordinates must contain either the same column letter or the same row number as the FROM coordinates. The VisiCalc program determines whether to move a row or a column by comparing FROM and TO coordinates: If the column letter in the two coordinates is the same, then a row is moved; if the row number is the same, then a column is moved.

The difference between the FROM and TO coordinates tells VisiCalc where to put the moved information. If the FROM coordinates (e.g., D5) have the same *column* letter as the TO coordinates (e.g., D3), then the contents of row 5 will move up to row 3. If the FROM coordinates (e.g., D5) have the same *row* number as the TO coordinates (e.g., B5), then the contents of column D will move left to column B.

VisiCalc makes room for the row or column you move by shifting the other rows and columns over. So moving a column or row to a new location does not "cover up" any other entries.

manual entry, and the remaining two problems appeared later in the manual after entries and problems pertaining to other commands. The amount of intervening material between the distributed problems ranged from one new manual entry and three nonpertinent problems to three manual entries and seven problems. To ensure for adequate spacing of the commands, we constructed a massed/distributed spacing template. Although the presentation order of commands was randomized for each subject, the template determined the relative positions of training problems within the manual.

Test Booklets. In addition to the training manual for each subject, a test booklet was constructed that contained the 12 problems that remained after the

Figure 4. A typical training problem, presented in (A) Guided Practice form and (B) Problem Solving form, which subjects saw in conjunction with an on-line VisiCalc spreadsheet, with contents as specified in (C).

A. Guided Practice Form

Alphabetize the names by putting the rows containing Steele and Stewart farther down in the appropriate spots. Start with cell A1 as the current cell.

TYPE THIS: /M . A7 [RETURN]
 /M . A7 [RETURN]

B. Problem-Solving Form

Alphabetize the names by putting the rows containing Steele and Stewart farther down in the appropriate spots.

Feedback, appearing on the following page:

You could have used the following sequence of commands (starting with cell A1 as the current cell) to solve the preceding problem.

 /M . A7 [RETURN]
 /M . A7 [RETURN]

C. Contents of VisiCalc Display in Initial Problem State

	A	B
1	Steele	Clerk
2	Stewart	Clerk II
3	Sanders	Manager
4	Schiff	Manager
5	Sebert	Accountant
6	Snyder	Sec'y
7	Sweet	Clerk III

practice problems were chosen for each command. The test problems were randomly ordered, and each problem appeared on a separate page of the test booklet. The problem-solving form of the problems was used, but no feedback was provided.

2.3. Subjects

Forty-four members of the Carnegie-Mellon University community (undergraduates, staff, and graduates) participated in the experiment. Subjects varied in previous computer experience from novice to experienced computer user. A questionnaire was used to rate subjects' experience and to screen out those who were familiar with electronic spreadsheet programs. (The rating of subjects' prior experience will be discussed further in Section 3.) Subjects were paid at a rate of \$3 per hour for participating in the experiment or received a combination of money and course credit for a psychology class.

2.4. Procedure

Subjects were run individually in two sessions: a training session and a testing session 2 days later.

Training Session. At the start of the training session, subjects were seated before an IBM Personal Computer (IBM-PC) displaying a blank VisiCalc spreadsheet. The training manual rested on a lecturn beside the IBM-PC. Subjects were instructed to read the manual one page at a time, without turning back to previous pages. Subjects in the experimental group were told that the manual would include training problems that they would solve on the computer and that they were permitted to type at the keyboard only while working on a problem. When the instructions for a training problem included the words "TYPE THIS," they were to type in the exact sequence of keys indicated. When the instructions did not provide a solution, they were to solve the problem on their own as efficiently as possible. Subjects in the control group were told that the manual contained descriptions of VisiCalc commands followed by problems exemplifying how to issue the commands. They were to study the examples, but not type anything at the keyboard. Subjects worked through the manual at their own pace; on the average, the experimental subjects took 1 ½ hr and the control subjects, 1 hr.

Two black-and-white video cameras were used to record the subjects' interactions with the computer and to collect reading and problem-solving times. One camera was focused on the manual and one on the screen of the IBM-PC. A mixer connected to a video cassette recorder produced a split image allowing us to record the top few lines of each manual page concurrently with the VisiCalc display on the screen. A millisecond timer was superimposed in the lower right corner of each frame. This allowed us to record and time the subject's interactions with the computer, as well as reading times per page of the manual.

Testing Session. Two days after the training session, subjects returned for the testing session in which they solved one problem for each command without feedback and without referring to the instructional text. The overall procedure for the testing session was identical to the training session; most subjects completed the test in less than 40 min. An experimenter was present during both sessions to call up the appropriate VisiCalc spreadsheet for each problem and to note the subject's success at working the problems.

3. RESULTS AND DISCUSSION

3.1. Scoring

We used several performance measures. One was success at solving the problems (accuracy). We awarded 1 point for each correct solution and 0 points for an incorrect solution. The correctness of a solution was judged by whether it

satisfied the goals specified for the problem, using appropriate commands. Partial credit (.5 point) was awarded if the subject attempted to use the appropriate command but missed some minor detail of the syntax. We also noted solution times for correct and partially correct responses. These were calculated from the timestamps on the frames of the videotape, measured from the first appearance of the page displaying the instructions for the problem to the time when the last command of the solution was entered. Although the data in Figures 6 and 8 report solution times in seconds, the analyses used $\log(\text{time})$ in order to normalize the data.

Subjects' previous computer experience was rated on the basis of a questionnaire. Subjects were asked to list the types of computers they had used, to list the programming languages they had studied and the duration of study, and to rate the frequency with which they used various computer applications (e.g., text editors, graphics packages, electronic spreadsheets, and statistical packages). Subjects who were familiar with electronic spreadsheets did not participate in the study. The remaining subjects were classified into three categories. They were rated as *experienced* computer users if they were familiar with more than two computer operating systems, had studied more than two programming languages for a total of at least one year, and were frequent users of text editors. *Intermediate* computer users were familiar with one or two computer systems, had one or two semesters of computer programming, and had used computers less frequently in general, as well as less frequently for text editing. Subjects with even less experience or no experience with computers were rated as *novices*.

The mean experience level of the control and experimental groups was computed using a 3-point rating scale: *novice* (0), *intermediate* (1), and *experienced* (2). Subjects in the experimental group had a mean rating of 1.0, and control subjects had a mean rating of 1.3. Experience was used as a grouping variable in all of the analyses to be reported. The results validated the ratings: Subjects with greater experience solved more problems correctly and tended to work more quickly. However, computer experience did not interact with any variables of interest. So, to simplify the exposition, the data to be reported are collapsed over experience levels.

The results also validated the categorization of the commands by difficulty: Difficult commands had significantly longer solution times at test than easy commands. However, Command Difficulty, like Computer Experience, did not interact with any other factors of interest. So we have also dropped the Difficulty factor from the discussion of this study.

The analyses for the experimental group are therefore the result of 3×2 analyses of variance (ANOVAs) over the Practice Form and Spacing factors. To compare the control and experimental groups, we collapsed the data from the experimental group over the Practice Form factor and performed a 2×2 ANOVA with Spacing (Massed vs. Distributed) and Group (Experimental vs. Control) factors.

3.2. Performance on Problems During Training

Accuracy. Figure 5 presents mean accuracy scores for the experimental group on the training problems as a function of practice form and spacing. Only data for the experimental group are reported for this measure, because control subjects merely studied the problems without interacting with the computer. There was a significant main effect of Practice Form on accuracy, $F(1, 27) = 19.9, p < .01$, with better performance in the guided practice condition than in the mixed and problem-solving conditions. Accuracy scores on guided practice problems were about 15 percentage points higher than on mixed practice and problem-solving problems; the latter two conditions did not differ significantly.

The spacing of the training problems in the training manual also influenced performance. Overall, subjects were 5% more accurate when problems were massed than when they were distributed, and this difference, though small, was reliable, $F(1, 27) = 4.3, p < .05$.

Solution Times. Similar results were obtained for solution times. Figure 6 presents the mean time (in seconds) that subjects took to correctly solve a training problem. Practice Form again produced a significant main effect, $F(2, 54) = 17.9, p < .01$. Subjects solved guided practice problems about 20 sec faster than mixed practice problems, $t(29) = 3.7, p < .01$, and about 30 sec faster than problem-solving problems, $t(29) = 5.0, p < .01$. Subjects were also faster at solving the mixed practice problems than the problem-solving problems, $t(29) = 2.9, p < .01$.

There was no main effect of spacing on solution times. Subjects appeared to take longer on massed problems than distributed problems in the problem-solving condition; however, the Spacing \times Practice Form interaction was not reliable.²

Figure 6 also lists the mean time that control subjects chose to spend studying a training problem and its solution. These times do not represent interactions with the computer, because subjects in this condition were not allowed to type at the keyboard. Rather, these data represent the mean amount of time that subjects studied the page of the manual containing a training problem, calculated from the timestamp when the page first appeared to the timestamp when the page was turned. It is interesting to note that the times for the control group are quite similar to the times in the guided practice condition. The similarity of the guided practice condition to the control group will be discussed further.

² The slower times in the massed problem-solving condition appear to be due to one especially difficult command, the "titles" command. Of the 10 subjects in this condition with mean solution times over 100 sec, 6 were working on the titles command (we did not counterbalance how often a particular command was assigned to a condition). Crucially, the titles command appeared in this condition a disproportionate number of times: 9 subjects saw it in the massed problems-solving condition, but only 2 saw it in the distributed problem-solving condition.

Figure 5. Mean accuracy scores for solving practice problems during training, as a function of form of practice and spacing.

Spacing	Form of Practice		
	Guided	Mixed	Problem-Solving
Massed	.99	.89	.84
Distributed	.98	.83	.78

Figure 6. Mean solution times (in seconds) for practice problems during training, as a function of form of practice and spacing. Times for the control group represent the time a subject spent studying a sample problem and its solution without typing at the keyboard at any time.

Spacing	Form of Practice			Control Group
	Guided	Mixed	Problem-Solving	
Massed	52.6	72.4	97.7	55.0
Distributed	52.5	71.2	88.3	56.5

Implications of Training Session Results. We are not inclined to give too much weight to the finding that guided practice produced the best results during training; in order to produce a correct solution to a guided practice problem, subjects simply had to follow the instructions. Carroll et al. (1985) observed that step-by-step instructions sometimes caused difficulties when subjects made mistakes or explored on their own. We did not observe subjects having difficulty following the guided practice instructions or getting back on track if they made an error. However, the presence of the experimenter may have led subjects to follow the instructions more carefully and discouraged them from exploring on their own.

The superior performance on massed problems during training is also to be expected. In the massed condition, the solution to a problem can be held in working memory and can serve as a model for subsequent similar problems. Therefore, it is of greater interest whether comparable results are obtained at test.

3.3. Performance on Problems at Test

Accuracy. Figure 7 presents the mean accuracy scores for test problems, as a function of practice form and spacing. Although practice form again produced a significant main effect, $F(2, 54) = 4.8, p < .05$, the source of the effect is quite different: The guided practice condition produced the worst performance at test rather than the best. Because accuracy was very similar in the mixed

Figure 7. Mean accuracy scores at test, as a function of form of practice and spacing.

Spacing	Form of Practice			Control Group
	Guided	Mixed	Problem-Solving	
Massed	.47	.65	.64	.49
Distributed	.59	.67	.71	.46

and problem-solving conditions, the effect of practice form is clearly due to the superiority of these two conditions over the guided practice condition.

Unlike performance during training, accuracy at test was not influenced by the spacing of the training problems, $F = 2.0$.

Solution Times. Figure 8 presents the mean solution times for correct responses. There was no main effect of Practice Form on solution times; subjects in all three conditions solved problems in about 90 sec. Taken together with the accuracy data, these results suggest that the lower accuracy of the subjects in the guided practice condition was due to a real difference in learning and not simply to a speed/accuracy trade-off.

There was a main effect of spacing on solution times, $F(1, 27) = 9.5, p < .01$. As shown in Figure 8, distributed spacing shortened overall solution times by an average of 24 sec. We had expected that increased problem-solving activity in the problem-solving and mixed practice conditions would reduce the benefit of distributed spacing as compared to the guided practice condition. Contrary to our expectations, though, the spacing effect appears to be the largest in the problem-solving and mixed practice conditions. However, the Practice Form \times Spacing interaction was not statistically reliable, $F = 1.9$.

3.4. The Control Group Versus the Experimental Group

Overall, performance of subjects in the experimental group was superior to that of the control group, those subjects who studied the training manual without interacting at all with the computer. Their data are given in the right-hand columns of Figures 7 and 8. By collapsing the data over the Practice Form factor, a 2×2 ANOVA was performed, using as factors of treatment Spacing (Massed vs. Distributed) and Group (Experimental vs. Control). The experimental group solved more test problems correctly, $F(1, 38) = 5.8, p < .01$, and was faster at solving the problems than the control group, $F(1, 36) = 8.6, p < .01$.³ Spacing produced no main effect on either measure. One might have ex-

³ The degrees of freedom differ in the two analyses because two control subjects failed to solve any test problems correctly. Data for these subjects could be included in the accuracy analysis but not the solution time analysis.

Figure 8. Mean solution times (in seconds) at test, as a function of form of practice and spacing.

Spacing	Form of Practice			Control Group
	Guided	Mixed	Problem-Solving	
Massed	89.1	110.2	106.6	134.6
Distributed	92.6	80.1	62.7	135.0

pected a Spacing \times Group interaction, because spacing had produced a main effect on solution times for the experimental subjects. However, the Spacing \times Group interaction was not reliable for either solution times, $F(1, 36) = 2.4$, $p > .1$, or accuracy, $F < 1$.

Although the performance of the experimental group was superior to the control group overall, the performance of the experimental subjects in the guided practice condition was very similar to that of the control group. As noted previously, subjects in the two conditions spent equal amounts of time studying the training problems. The two conditions also produced similar levels of accuracy at test. However, there appears to be an advantage for guided practice training over simply reading, in terms of solution times at test: The control group spent 135 sec per problem, whereas the mean solution time in the guided practice condition was only 90 sec. The faster time in the guided practice condition may be due to the hands-on, step-by-step interaction with the computer that this condition provided. However, this is not the only possible explanation. Subjects in the guided practice condition were part of a within-subject design and thus were also exposed to problem-solving training for other commands. This problem-solving practice may have contributed to faster times overall. To resolve this point, we would have to use a between-subjects design, to see whether subjects trained exclusively with guided practice would still produce faster times than the control group.

3.5. The Disadvantages of Guided Practice Training

Why was guided practice training less effective than training that included problem solving? One possible explanation involves the amount of time subjects spent on the training problems. Subjects spent the least amount of training time on problems in the guided practice condition and the most time on problem-solving problems. One could argue that the form of practice is irrelevant except as it motivates subjects to spend more time on training, thereby producing a stronger representation of the procedures in memory.

Two features of the results are inconsistent with this interpretation. First, even when the time spent on training was equivalent, the type of training still

influenced performance. In particular, the control group put in slightly more study time per problem than subjects in the guided practice condition, but the guided practice condition produced significantly shorter solution times at test. Second, differences in study time did not always produce corresponding differences in performance. Subjects spent significantly less time during training on mixed practice problems than on problem-solving problems. However, at test the two conditions did not differ significantly in terms of either accuracy or solution times. Of course, the shorter training times for the mixed condition were probably due to quick responses to the first mixed trial which was always in guided practice form and did not require problem solving.

In order to further refute the argument that our results are confounded with total training time, we computed correlations between training time and test performance for each of the three types of training. The correlations between training time and test accuracy tended to be small and negative (guided practice condition, $r = -.02$; problem-solving condition, $r = -.29$; mixed practice condition, $r = -.12$), suggesting, if anything, that subjects who spent more time on training problems tended to perform less accurately at test. The correlations between training time and solution times at test tended to be small and positive (guided practice, $r = .22$; problem solving, $r = .21$; mixed practice, $r = .13$), which may simply reflect differences in typing speed among the subjects.

Thus, although the evidence is indirect, there seems to be sufficient reason to believe that the difference between the guided practice condition and the conditions that include problem solving are due to the quality of the activities that subjects perform during training and not simply the quantity of time and attention they devote to training.

3.6. Problem Solving Versus Mixed Practice

Is mixed practice just a mixture? We originally included this condition because we suspected that subjects would need an example of a correct command in order to interpret the abstract syntactic rule (Charney & Reder, in press; Reder et al., 1986). We expected subjects to benefit from seeing one guided practice problem before trying problems on their own. However, the results reveal very little difference between the problem-solving and mixed conditions. During training, solution times were shorter in the mixed condition; however, this was probably due to the combination of a fast guided practice problem followed by two slower problem-solving problems. Thus our subjects did not appear to benefit from working through an example before attempting independent solutions.

4. CONCLUSION

4.1. Discovery Learning and Learning by Example Revisited

We began this article by noting that the discovery learning strategy leads to different principles for manual design than learning by example. The results of this experiment force us to reconsider both strategies.

With respect to learning by example, the results reported here are inconsistent with those of Sweller and Cooper (1985), who advocated the study of worked-out examples over pure problem solving for learning a cognitive skill. Although the training paradigms that we used differed somewhat from Sweller and Cooper's, the comparison is still apt. We found that performance at test was consistently better when training consisted of problem solving than when it presented worked-out examples. Even when examples were combined with problem solving, performance at test was not significantly better than with pure problem solving. The difference between our results and Sweller and Cooper's may be explained in two ways. First, the difference might be attributable to the delay we imposed between training and the test. Sweller and Cooper's subjects were always tested immediately after the training session, whereas our subjects were tested after a 2-day delay. Conceivably, it is easier and more effective to use an example as a model than to generate a solution to a problem, but this advantage disappears with time as examples become increasingly difficult to retrieve from memory. This would suggest that when the test immediately follows training, subjects would be able to retrieve the example and perform better than those who only have trained with problem solving. However, after a delay, the example may no longer be retrievable. In this case, all subjects are forced to make use of problem-solving techniques and those who have had practice applying the procedures with problem solving will perform better.

The second explanation has to do with the demands of the tasks. Sweller and Cooper (1985) were dealing with algebra, where the basic mathematical operations were already well learned by the subjects and where distinct problem types may be easily isolated. Studying examples may have been more advantageous than problem solving in this situation because the major challenge is learning to recognize the problem types. In contrast, learning to select and execute computer commands requires learning new operations as well as recognizing the situations in which they are appropriate. In this case, studying examples, or even working through step-by-step instructions, is apparently less useful than generating commands independently.

The relatively poor performance in the guided practice condition is consistent with Carroll et al.'s (1985) finding that step-by-step tutorial instruction is inadequate. However, our results suggest that the goal-setting and exploration

aspects of the discovery learning strategy, as employed in Carroll et al.'s guided exploration materials, may not be the primary source of the advantage they found over training with the commercial tutorial manual. A substantial part of the advantage may be due simply to the problem-solving activity their subjects engaged in. Further research that directly contrasts problem solving and discovery learning will be necessary to determine whether goal-setting and exploration activities confer additional benefits over and above problem solving.

4.2. Speculations and Implications

We note that our results are limited to problems that can be solved by applying a single method rather than requiring combinations of procedures. It is interesting to speculate on whether our results will extend to more complex problems. We noticed that subjects working on training problems in the problem-solving condition would sometimes combine procedures in inefficient or ineffective ways. After finishing their attempt, they would closely consider the solution provided on the feedback page and draw conclusions about why it might be better. We suspect that subjects in the guided practice condition were less capable of evaluating the supplied solutions, because they were not forced to develop an independent solution first. We expect that if subjects are presented with complex problems that allow several solutions of varying efficiency, they will again learn more sophisticated solution strategies through problem solving with feedback than through studying examples or following guided practice problems.

It is also interesting to speculate on the generality of these findings for different subject populations. We argued in the introduction that discovery learning may be ineffective for people who are not familiar enough with typical problem situations in a domain to set reasonable goals. What if the subjects were more familiar with some aspect of the domain? Carroll et al.'s (1985) subjects were secretaries learning to use a word processor. They were inexperienced computer users, but highly experienced at the types of tasks they would use the computer for (i.e., typing letters and manuscripts). In this case, the subjects may have been able to invent realistic problem situations for themselves that focused their exploration on the most relevant features of the system and helped them invent plausible problems to try to solve. In contrast, our subjects were learning a new computer application without any real knowledge of the uses to which a spreadsheet might be put. As a result, they may have been completely lost if left to explore the spreadsheet on their own: The training problems clarified the goals to which the procedures might be applied as well as the details of actually executing the procedures. This may be why we found no interactions between previous computer experience and training condition in our study; although the experienced users were better at learning to execute the commands effi-

ciently, both groups of subjects had to learn about the situations in which a command was appropriate.

In any case, the results of this study do not bode well for interactive tutorial manuals in their present form, because most such manuals rely exclusively on guided practice. Novice users working through such tutorials may successfully complete all of the exercises and still not learn what they need in order to use the commands on their own. We believe that adapting the problem-solving form of training developed for this study and distributing practice will produce much more effective user manuals.

Acknowledgments. We thank Gail Wells for help with all phases of this research and Bonnie John for facilitating our access to the User Studies Lab.

Support. The work reported here was sponsored by Office of Naval Research Contract N00014-84-K-0063 and in part by National Science Foundation Grant BNS-03711 to Lynne Reder.

REFERENCES

- Carroll, J. (1984, August). Designing MINIMALIST training materials. *Datamation*, pp. 125-136.
- Carroll, J., Mack, R., Lewis, C., Grischowsky, N., & Robertson, S. (1985). Exploring exploring a word processor. *Human-Computer Interaction*, 1, 283-307.
- Charney, D., & Reder, L. (in press). Initial skill learning: An analysis of how elaborations facilitate the three components. In P. E. Morris (Ed.), *Modelling cognition*. Chichester, England: Wiley.
- Draper, S. (1984). *The nature of expertise in UNIX* (Tech. Rep. No. 8402). San Diego: University of California, Institute for Cognitive Science.
- Glenberg, A. (1979). Component-levels theory of the effects of spacing of repetitions on recall and recognition. *Memory & Cognition*, 7, 95-112.
- Hintzman, D. (1976). Repetition and memory. In G. Bower (Ed.), *The psychology of learning and motivation* (pp. 47-91). New York: Academic.
- Price, J. (1984). *How to write a computer manual: A handbook of software documentation*. Menlo Park, CA: Benjamin/Cummings.
- Reder, L., Charney, D., & Morgan, K. (1986). The role of elaborations in learning a skill from an instructional text. *Memory & Cognition*, 14, 64-78.
- Scharer, L. (1983, July). User training: Less is more. *Datamation*, pp. 175-182.
- Sweller, J., & Cooper, G. (1985). The use of worked examples as a substitute for problem solving in learning algebra. *Cognition and Instruction*, 2, 59-89.
- Tausworthe, R. (1979). *Standardized development of computer software, Part II: Standards*. Englewood Cliffs, NJ: Prentice-Hall.

HCI Editorial Record. First manuscript received May 29, 1986. Revision received December 12, 1986. Accepted by John Black. — Editor
