

The role of elaborations in learning a skill from an instructional text

LYNNE M. REDER, DAVIDA H. CHARNEY, and KIM I. MORGAN
Carnegie-Mellon University, Pittsburgh, Pennsylvania

In these studies, we examined the role of elaborations for subjects learning a procedural skill (viz., using a personal computer) from an instructional text. In Experiment 1, we compared two sources of elaborations: those provided by the author and those generated by learners while reading. In the latter condition, subjects were given advance information about the tasks they were to perform so that they would generate more specific task-related elaborations while reading. Each source of elaborations facilitated skill performance. This result contrasts with results of the past experiments testing declarative knowledge in which author-provided elaborations were found to hurt performance. In Experiment 2, the author-provided elaborations were classified into those illustrating the syntax of the operating system commands and those explaining basic concepts and their applicability. Syntax elaborations produced significant facilitation for experienced and novice computer users. Concept elaborations produced no reliable improvement.

An important question to both memory theorists and pedagogists is, "What variables will improve the learning and retention of written information?" One such variable that has been the topic of considerable speculation and research is the effect of elaborations (J. R. Anderson & Reder, 1979; Bransford, 1979; Chiesi, Spilich, & Voss, 1979; Craik & Tulving, 1975; Mandl & Ballstaedt, 1981; Mandl, Schnotz, & Tergan, 1984; Reder, 1976, 1979, in press; Weinstein, 1978). In the view of most researchers, there are several reasons why elaborations should help subjects learn and remember the main ideas of a text. Elaborations provide multiple retrieval routes to the essential information by creating more connections to the learner's prior knowledge. If one set of connections is forgotten, it may be possible to retrieve the desired information another way. Furthermore, if the learner forgets an important point, it may be possible to reconstruct it from the information that is still available.

Elaborations can arise from two distinct sources: first, the text itself can contain elaborations of the main ideas, and second, the reader can generate them independently while reading. We use the same term for both types, because we define elaborations as any information that supports, clarifies, or further specifies the main points of a text. Elaborations can take many forms, including examples, details, analogies, restatements, and deductions. There are merits and drawbacks both to the elaborations generated by the reader and to those provided by the author. Elaborations provided by the author of the text may be more accurate than those the reader devises, since

the author is presumably more knowledgeable about the topic. On the other hand, the reader's own elaborations are likely to be more relevant to his or her immediate purpose for reading.

There has been ample research supporting the idea that reader-generated elaborations facilitate retention. This support comes from experiments in which subjects had additional knowledge that allowed them to generate more elaborations than other subjects generated. Some experiments contrasted subjects who had a good amount of domain-relevant knowledge (for example, knowledge about baseball) with subjects who had little relevant knowledge (e.g., Arkes & Freedman, 1984; Chiesi et al., 1979). In other experiments, some subjects were provided with additional information that was relevant to a passage to be read (e.g., Bower, Black, & Turner, 1979; Brown, Smiley, Day, Townsend, & Lawton, 1977; Sulin & Dooling, 1974). In all cases, subjects who had access to more relevant knowledge were more likely to intrude non-presented, but relevant, information and were also more likely to false alarm to plausible inferences based on this additional information. Therefore, it is reasonable to conclude that these subjects, in fact, were elaborating on the presented material with their relevant prior knowledge.¹ More important from our perspective, the subjects in these studies also showed significantly better retention of the gist of the material and better understanding of it (R. C. Anderson & Pichert, 1977; Arkes & Freedman, 1984; Bartlett, 1932; Bower, 1976; Brown et al., 1977; Dooling & Cristiaansen, 1977; Owens & Bower, 1977; Schallert, 1976; Weinstein, 1978).

Although reader-generated elaborations have been found to facilitate retention, the evidence concerning author-provided elaborations indicates that the latter do not provide similar benefits. Author-provided elaborations may, in fact, impair retention of the central ideas, as compared to studying these ideas in isolation. In 10 studies,

The work reported here was sponsored by the Office of Naval Research Contract No. N00014-84-K-0063, and in part by the National Science Foundation Grant No. BNS-03711. Requests for reprints should be addressed to L. Reder, Department of Psychology, Carnegie-Mellon University, Pittsburgh, PA 15213.

Reder and J. R. Anderson (1980, 1982) found that students who read fully elaborated chapters, taken verbatim from standard college textbooks, consistently performed worse than did students who read chapter summaries that were one fifth as long. The advantage for the summaries (which appeared in both reaction time and percentage correct measures) held up at a variety of retention intervals (ranging from 20 min to 1 year), and it held for various tests of declarative memory, including forced-choice verification, short answer, and free recall (Allwood, Wikstrom, & Reder, 1982). Performance was better even on new material when related material had previously been studied in summary form. The advantage for summaries was also found under a variety of study conditions. In the initial experiments, a fixed study time was imposed on subjects in both text conditions. However, Reder (1982) also found an advantage for summaries in a nonlaboratory setting, in which subjects studied the materials at home at their own paces. Furthermore, Reder and J. R. Anderson (1982) equated reading time for each presented sentence in both conditions. That is, the main points were presented for equal amounts of time in isolation on a computer screen; the subjects in the elaborated condition took additional time to study elaborations after each main idea. Performance in the summary condition was superior.²

Not all research on author-provided elaborations has found that elaborations impair learning; however, the conditions under which such elaborations benefit the learner tend to be rather specialized. Stein and Bransford (1979) studied subjects' recall of an adjective cued by the sentence frame within which it had been studied. The elaborations in these cases were additional phrases or clauses that increased the importance of the adjective to the plausibility of the sentence. Mandl et al. (1984) found that elaborated texts facilitated recall and comprehension, but only when the reader was very knowledgeable in the topic area; otherwise, elaborated texts produced worse performance than unelaborated ones. Rothkopf and Billington (1983) found that elaborated passages were sometimes superior to unelaborated passages, but only when they were mixed into the same texts: when one text was uniformly elaborated and another was summarized, the summarized version produced better retention of the central points. Bradshaw and J. R. Anderson (1982) attempted to devise especially related elaborations that would facilitate recall of an entire sentence (cued by the sentence subject). For a fixed amount of study time, the best they could do was to get equivalent learning performance in the elaborated condition and in the isolated (unelaborated) condition.

Overall, the pervasive finding seems to be that, especially with a fixed study time, people learn facts best when they study those facts without reading elaborations. Yet, the finding that author-provided elaborations are often ineffectual and sometimes even detrimental to learning is a serious and curious charge. The implications for textbook production would be grave if one actually believed

the result. One factor that we believe is crucial to the effectiveness of the elaborations found in instructional texts is the kind of learning that is expected to take place. Indeed, educators and laymen alike often assert that it is much less important to know a set of facts than to know how to use these facts. A similar distinction between fact learning and skill learning has received considerable attention in recent years in the cognitive science literature, under the label of declarative versus procedural learning (e.g., J. R. Anderson, 1980; Posner, 1973; Shiffrin & Schneider, 1977). Below, we briefly contrast the nature of each kind of learning and explain why we expect author-provided elaborations to play a different role in each.

Why Do Author-Provided Elaborations Impede Fact Learning?

There are several characteristics of tests of factual knowledge that may lead a person to perform better after studying a summary than after studying a full, elaborated text. First, tests that ask subjects to recall or recognize studied statements require retrieval of specific facts learned at a particular time. The stronger the trace is, the more likely that it will be retrieved at test. A proposition or fact is strengthened in memory to the extent that the subject devotes more attention to it. The studying of summaries facilitates performance on tests of factual knowledge, because it allows the reader to devote full attention to the essential facts, exactly those that must be retrieved at test. In other words, studying elaborated texts impedes learning the main points of the text because reading the elaborations reduces the amount of time subjects can devote to the main points. This Total Time Law is a well established verbal learning phenomenon (e.g., Bugelski, 1962; Cooper & Pantle, 1967).

The Total Time Law cannot completely explain the advantage of the unelaborated versions, however. When study time was equated for main points, but one condition also gave subjects additional time to study related facts after each main idea, performance was still significantly worse with the additional facts (Reder & J. R. Anderson, 1982). An obvious explanation for the result is that elaborations can also interfere with recall. Interference differs from the total time explanation in that it affects retrieval rather than encoding. There is ample evidence for the existence of retrieval interference both on recall (e.g., Postman, 1971; Postman & Stark, 1969) and on response times to verification (e.g., J. R. Anderson, 1974; Reder & J. R. Anderson, 1980).

Another reason that elaborations do not help performance in factual tests is that there is little uncertainty in the testing situation about how to apply the knowledge. In a recognition or a recall test, it is usually clear what information is needed, when or why to retrieve it, and what to do with it once it is retrieved. The information found in an elaboration is seldom called on in such tests

and therefore only distracts the reader, when his or her time would be better spent studying the targeted facts themselves.³

Why Might Author-Provided Elaborations Facilitate Skill Learning?

Our conjecture is that tests of fact learning place specific demands on the retrieval processes and that studying summaries focuses subjects' attention in just the right places to produce the best performance. This analysis suggests that author-provided elaborations can facilitate performance if the tests no longer place such a high premium on retrieving only the main points. When subjects must use the information, in the sense of determining appropriate contexts and methods of application, then we expect the benefit of the additional information for performing the tasks to outweigh the liability of having less study time for the main points and additional retrieval competition.

The task in our studies involves the acquisition of a cognitive skill (viz., learning to manipulate files and directories on a personal computer). In the task, a small set of basic procedures can be applied to a variety of novel situations. Good skill performance on a novel task requires three things: (1) Appreciating the meaning of concepts novel to the new skill domain. For example, in learning to use an IBM Personal Computer, one might be introduced for the first time to function keys that allow the user to reexecute a command that was already issued, without retyping it. (2) Remembering and comparing procedures to select the most appropriate one for the situation at hand. Knowing at some level that such a function key exists does not mean that the user will remember to use it. (3) Remembering the exact syntactic form for the procedure and how to apply it in a specific situation. A user might remember that the typing step could be saved by using the special key but be uncertain as to which of the 10 function keys performs that function and exactly how to apply it (e.g., whether the carriage return key must be pressed afterward, or the escape key).

Elaborations in the text may touch on any of these topics: what the basic concepts are, when they are relevant, and how one applies them. For example, when the goal of a task does not exactly match the function of any known procedure, elaborations may provide a subject with deeper understanding of the function of each individual procedure and thus facilitate construction of an effective combination of procedures. Elaborations about what conditions affect the usefulness of a procedure can help subjects plan out more efficient sequences of actions. Finally, supplementing a general syntactic rule for a computer command with specific examples can help subjects set more specific standards for what their own commands must look like.

Elaborations seem most important for subjects learning to select and execute procedures correctly. There is some experimental evidence that examples facilitate the selection and execution of procedures in other skill-learning situations. Ross (1984) found that subjects chose between equivalent computer text-editing procedures on

the basis of the superficial similarity between the nominal situation at test and the situation evoked in instructional examples. Pepper (1981) also found that students attended closely to examples. Those who read a carefully written computer programming chapter that included numerous examples not only rated it more highly than comparable chapters without examples, but also answered more programming problems correctly than did students who read the other chapters.

In Experiment 1, we crossed the availability of two sources of elaboration on skill performance: whether elaborations were present in the instructional manual and whether readers received information about the tasks before they read the manual. Providing prior knowledge of the test material has been explored extensively in tests of declarative memory (see R. C. Anderson & Biddle, 1975; Reder, 1982, in press; and Rickards, 1979, for reviews). Readers perform better on tests of factual knowledge when they are given the questions prior to studying the material; however, if they are given prior knowledge of only some of the questions, they perform poorly on the unexpected test questions. In a skill-learning domain such as this, we expect prior knowledge of the task to facilitate performance by focusing the reader's attention and stimulating the generation of task-specific elaborations.

In the fact-learning literature, there is evidence that a subject's own elaborations are better retained than those provided by the experimenter (e.g., Bobrow & Bower, 1969; Rohwer & Ammon, 1971; Rohwer, Lynch, Levin, & Suzuki, 1967; Rohwer, Lynch, Suzuki, & Levin, 1967). Nonetheless, we suspect that in a skill-learning domain, reader-generated elaborations will be less reliable than those the author can provide. Thus, although we expect reader-generated elaborations to improve subjects' performance, relative to having no source of elaborations available at all, it is unclear how they will compare to author-provided elaborations and how they will interact.

EXPERIMENT 1

In Experiment 1, subjects were asked to read one of two versions of a user's manual for the IBM Personal Computer (IBM-PC). We then measured their facility at using the computer without the manual to perform a specific set of tasks.

Method

Design. The experiment used a 2x2 between-subjects factorial design, where the first variable determined whether the document contained elaborations or not, and the second determined whether subjects read the task instructions prior to studying the document or not. Subjects were randomly assigned to conditions, with the constraint that they be evenly divided among the four conditions. Although we noted how much previous computer experience our subjects had, we did not control for this variable in assignment to conditions, except to ensure that no subject had ever worked on a microcomputer. Instead, we used prior experience as a covariate in our data analyses.

This experiment actually contained two experiments, the second a virtual replication of the first. Any differences in materials or

procedures will be noted, as will the rationale for any modifications in the replication.

Materials. Two versions of a user's manual were developed for teaching novices to use the disk operating system (DOS) on an IBM-PC. Both versions were constructed by modifying portions of the official IBM (Microsoft, Inc., 1983) documentation. The same basic information was presented in both versions, which differed only in the degree of elaboration of concepts and procedures for using the described commands. The manuals were divided into two sections. The first section discussed concepts underlying the IBM-PC and its operating system, such as disk drives, directories and sub-directories, and use of wild-card characters. This section laid the groundwork for the specific commands discussed in the second section, since the command syntax requires the location of an object (such as a file or a directory) to be specified in terms of a disk drive and a path through a hierarchy of directories and subdirectories. The second section of the documentation introduced 11 DOS commands (a complete list is provided in Appendix A). The manual described what each command does, what parameters must be specified when it is issued, any optional parameters, and any other special information, such as how the computer interprets wild cards in the context of that command.

The unelaborated version of the document was constructed by deleting portions of the elaborated manual, such as examples, analogies, metastatements, and definitions. Appendix B shows samples from the elaborated and unelaborated manuals for each of these types of elaboration. None of the elaborations contained any new information necessary to completing the criterion tasks. The elaborated version contained more than twice as many words as the unelaborated version. (In the first run of the experiment, the elaborated version contained 11,216 words and the unelaborated version contained 5,011; in the replication experiment, the manuals were trimmed to 10,605 and 3,542 words, respectively.)

Apparatus. Two IBM-PCs, each with two disk drives, were connected by a cable between their serial ports. Software was developed to record the subject's interactions with the computer. While the subject issued commands at one IBM-PC, the commands and the computer's responses were echoed across the cable and recorded (with a time stamp) in a file on the second IBM-PC, which was screened from the subject's view by a room divider.

Subjects. Eighty-eight members of the Carnegie-Mellon University (C-MU) community (students, faculty, and staff members) participated in the experiment, 45 in the first replication, and 43 in the second. All subjects were occasional to frequent users of C-MU's DEC-20 computer system, running the TOPS operating system (a system which does not support subdirectories). No subject had previously worked on a microcomputer. Three subjects were nonnative speakers of English, but fluent enough to completely understand the documentation. Subjects received either money or class participation credit or a combination of the two. (In the replication, only subjects who had taken no more than one computer programming course were allowed to participate. Initially we paid \$4/h for participation but lowered our rate to \$3/h for the replication.)

Procedure. Experimental sessions consisted of a reading period followed by a task performance period. At the outset of the reading period, half the subjects (the before group) were given printed instructions describing the criterion tasks, which they read before reading a version of the manual. The other subjects (the after group) did not see these instructions until just before they performed the tasks. Within the before and after groups, half the subjects read the elaborated manual and half read the unelaborated version. All subjects were told to review the important points as often as time permitted, because the manual would not be available while they performed the tasks. Subjects studied the manual while seated in front of the IBM-PC. They were told to examine the keys referred to in the documentation, but not to issue any commands, touch the machine, or take notes. When the reading period was over, the ex-

perimenter returned to remove the manual and begin the task performance phase.

Below are the task instructions used in the replication:

These tasks will allow you to practice using the concepts you learned from the manual. You may work on these tasks in any order. Continue working until you are satisfied that you have completed the tasks to the best of your abilities. We want you, however, to work as efficiently as possible.

Task 1. Before you, in drive A, is a diskette containing a number of files. Some of these files have the word "PART" in their names, such as file "PART.1." We want you to change the names of these files. The new name that you should give each file appears as the first line of that file. So, inspect the contents of each file that now has "PART" in its name and give the file the name that you find in the first line of the file.

Task 2. Four of the files on the diskette have the word "DATA" in their names, and the abbreviation of a month in their extension, such as DATA.MAR. We want you to create a fifth data file named ALLDATA.83 that contains the contents of the other four data files appended together. Within ALLDATA.83, the files should appear in "chronological" order; that is, the contents of DATA.MAR should precede the contents of DATA.JUN because March is earlier in the year than June.

Task 3. Next, you should create two subdirectories on the diskette in drive B. One subdirectory is to be named PROGRAMS and the other named DATA. Move all the files that have the word "Program" in their names from drive A into the PROGRAMS directory on drive B. And, similarly, move the "Data" files (including ALLDATA.83 from Task 2, if you have already created it), into the DATA directory. You do not want any Program or Data files to remain on the diskette in drive A.

Task 4. Finally, you should eliminate the SOURCE directory and everything it contains from the root directory of the diskette in drive A. The root directory on drive A should now contain only a list of files.

Your task is complete.

Even though the computer recorded all interactions between the subject and computer, the experimenter was present to determine whether the subject had arrived at an impasse (i.e., could not complete part or all of a task). The subject either gave up or was stopped after approximately 10 min of fruitless effort. At this point, the experimenter replaced the diskette that the subject was working on with a prepared diskette, on which the procedures for that part of the task had already been completed. In this way, the subject could proceed to the next part of the task as if he or she had actually completed the problematic part. Subjects were not allowed to ask questions, unless the question was of a superficial nature, such as, "Which key is the carriage return?" (This was not immediately obvious on the IBM-PC keyboard.) The entire experiment took approximately 1.5 h.

A few differences in procedure between the two replications are worth noting: (1) The first 40 subjects were given a fixed 45 min to study the manual; the second group of subjects were given up to 60 min but were permitted to stop sooner if they wanted to (study times were collected). In both cases, subjects were told when half the maximum study time had elapsed. (We permitted additional reading time in the replication because we felt that subjects in some conditions did not have enough time to study the manual when given only 45 min.) (2) In the replication, subjects in the before group were allowed to keep the task instructions while reading the manual. Initially, the before group returned the instructions after reading them once, and saw them for the second time when the task performance phase started (i.e., when subjects in the after group first received them). Allowing the subjects to keep the instructions was intended to eliminate the additional memory load of the task in-

structions while the subjects read the manual and to minimize the possibility of misremembering the tasks while studying the manual. (3) The task requirements in the replication differed in three ways from those used for the first version of the experiment. First, Task 2 was added in the replication to provide broader coverage of the commands in the manual. Second, Tasks 1 and 3 were made less repetitive (e.g., subjects renamed 6 files in the replication as opposed to 15 files). Finally, Task 4 was modified to force subjects to draw more on what they learned about subdirectories. Subjects were now forced to specify paths to deeply embedded subdirectories in their commands or change the default directory in order to successfully complete the task.

Results

The data of 8 subjects were discarded (leaving exactly 40 per replication), 5 because of computer failure, 2 because the subjects were so inexperienced with computers that they refused to continue the experiment shortly after beginning the criterion task, and 1 because his protocol revealed that he had worked on the tasks for several minutes during the study period.

In the replication, subjects were allowed to study the manual for up to 1 h (rather than for a fixed 45 min). The mean reading time was 49 min for the elaborated-before condition, 48 min for elaborated-after, 45 min for unelaborated-before, and 40 min for the unelaborated-after condition. The differences in reading times as a function of condition were not significant; however, 60% of the subjects in the elaborated-before condition studied the manual for over 50 min, whereas only 10% in the unelaborated-after condition did so.

In the analyses reported below, we always included the factor of replication. There were neither significant effects due to replication for any of the dependent measures, nor significant interactions of replication with any of the factors. Therefore, in our discussion of the results, we ignore the replication factor.

The protocol of a subject's interactions with the computer was stored in a file, which was subsequently analyzed by means of a computer program. The program allowed the experimenter to partition the protocols

according to which task the subject was working on. The program counted commands (e.g., how often a subject issued the TYPE command while working on Task 1) and calculated time intervals, both within and across task partitions. The partitioning of the protocols was carried out independently by two judges for a random subset of the data. The agreement between the judges was quite high ($r = .98$); any disagreements were resolved to mutual satisfaction.

There are many conceivable measures of performance. Rather than attempting to report them all (consult Reder, Charney, & Morgan, 1984, for a more thorough summary), we have selected a few representative measures: percentage of tasks completed, time spent on task, number of commands issued to perform the tasks, and performance efficiency for completed tasks only. Efficiency was defined as performance relative to the minimum number of commands required. These measures are presented in Table 1 as a function of whether each source of elaborations was available.

A consistent pattern emerges with all measures of performance, although more strongly in some than in others. Performance is better if subjects have an elaborated representation of the material; however, one source of elaboration is sufficient, and it does not matter whether it is author-provided or subject-generated. Another way to state this is that the unelaborated manual in the condition in which subjects did not have prior knowledge of the task produced the worst performance by all measures.

The performance measure that produced the smallest effect was the proportion of tasks completed correctly (the top row of Table 1). Each task was scored individually for a subject, with partial credit awarded for correctly completing any part of a task. These data represent the mean scores over all tasks attempted. There are no significant effects of either experimental variable on the ability to finish the assigned tasks, although performance in the condition with no source of elaborations appears to be slightly worse. (In the first run of the experiment, per-

Table 1
Performance on Task as a Function of When Task Instructions
Were Given and Type of Manual Studied, Experiment 1

	Before		After	
	Elaborated Manual	Unelaborated Manual	Elaborated Manual	Unelaborated Manual
Proportion of Tasks Correctly Completed	.80	.80	.85	.76
Mean Time on Task in Minutes (All Tasks)	33.5	36.1	29.4	40.2
Mean Number of Commands Issued (All Tasks)	95.8	94.2	76.8	101.8
Proportion of Commands Issued Per Minimum Step Required (Completed Tasks Only)	2.88	2.34	2.29	3.23

formance ranged from 75% to 93%. The elaborated manual in the after-instructions condition produced significantly better performance than did the other three conditions: however, when study time was allowed to deviate up or down from the enforced 45 min, performance of subjects in all conditions averaged about 80%. We take this relatively high level of performance as evidence that even our minimal (unelaborated) manual adequately explained the concepts and procedures so that all subjects could perform the tasks competently, at least when no time restrictions were imposed on task completion.

There were significant effects of elaborations on the quality of the performance by such measures as the time taken on the tasks and the number of commands issued. Rows 2 and 3 of Table 1 present these data. Subjects were significantly faster to perform the tasks if they had read the elaborated manual [31 min vs. 38 min; $F(1,72) = 5.6$, $MSe = 161$, $p < .01$]. Subjects in the elaborated-after condition issued significantly fewer commands than did subjects in the other three conditions [$t(72) = 3.5$, $SE = 5.9$, $p < .01$].

The time and steps measures just reported include data from tasks which subjects failed to successfully complete (i.e., they received no credit or partial credit for their work on the tasks). The scores for the incomplete tasks might skew these measures: the scores could be artificially high if subjects persisted in working on a task without success, or artificially low if they gave up immediately when they did not think they could do it. Therefore, the number of commands a given subject issued for a completed task was compared with the minimum number required for that task. The ratio of actual steps to minimum required are given in Row 4 of Table 1. We call this measure "efficiency."

There was a significant interaction of the extent of elaboration with whether subjects had prior knowledge of the task on efficiency [$F(1,72) = 6.82$, $MSe = 1.6$, $p < .05$]. Performance in the elaborated-after condition was superior to the other three conditions [$t(72) = 2.2$, $SE = .24$, $p < .05$]. The unelaborated manual produced somewhat better performance than did the elaborated manual when subjects had prior knowledge of the required tasks (the before conditions); however, the interaction is due primarily to the worst condition (unelaborated-after), in which subjects read the unelaborated manual without prior knowledge of the tasks. No contrast involving the elaborated-before condition by itself was significant.

Discussion

In the introduction, we noted that both the Total Time Law and the phenomenon of retrieval interference would predict that author-provided elaborations should hurt the acquisition and retrieval of central ideas from a text. Since the findings that such elaborations hurt performance were based exclusively on declarative knowledge tests (recall or recognition), we hypothesized that in a skill-learning domain, studying elaborations would help a learner de-

side which procedure to apply to solve a given problem and how to apply general rules in specific situations.

It would be useful at this point to summarize our findings: Subjects in all conditions managed to complete roughly the same proportion of the tasks; however, subjects who studied the elaborated manual worked more efficiently than subjects who studied the unelaborated manual, in terms of both the time they spent on tasks and the number of commands they issued. Subjects consistently performed poorly in the unelaborated-after condition, where no source of elaborations was available. This result is in sharp contrast to the results of Reder and J. R. Anderson (1980, 1982), for example, whose subjects always performed at least as well with the unelaborated version of a text. Thus, there is support for the hypothesis that, unlike in the domain of declarative fact learning, studying elaborations in a text does help people perform a skill.

One area of past research reviewed earlier did find that elaborations improve memory for main points: recall was better when people could use prior knowledge about a topic to generate their own elaborations while they read. In these studies, also, we found some evidence that having a specific task or problem in mind while reading a manual helped people apply what they were reading to their tasks. However, our results suggest that when people already know what tasks they must perform, they benefit very little from seeing examples and other elaborations in the text. The reader can process the manual selectively and generate his or her own task-specific elaborations. The elaborations in the manual are less relevant to the specific tasks and appear to distract the reader from this critical process. Although subjects who did not have the tasks in mind as they read were much better off with the elaborated version of the manual, subjects who knew about the tasks in advance performed just as well with the shorter, unelaborated version of the manual; this suggests that one source of elaborations is sufficient (i.e., performance is not better with multiple sources of elaborations).

It is interesting to speculate about why performance in the elaborated-after condition was often significantly better than performance in the other three conditions: in particular, why having both sources of elaborations available in the elaborated-before condition did not boost performance above that of either source alone. We suspect that, in part, it is unnecessary to have two sources of elaborations and, in part, having both sources available makes it more difficult to exploit either one. It was too demanding in the elaborated-before condition to process the author-provided elaborations, keep in mind the task requirements, and generate task-specific elaborations. The reading time data in the replication provides some evidence for this interpretation: 60% of the subjects in the elaborated-before condition studied the manual for over 50 min, whereas only 30% of those in the elaborated-after condition did so. However, having more time to study

the manual in the replication did not improve the relative performance of this condition.

How does performance in the elaborated-after condition compare to that in the unelaborated-before condition? Although the cell means for the elaborated-after condition are better on all measures, the differences in most cases are not very large and do not differ reliably from the unelaborated-before measures. (Typically, the elaborated-after condition was significantly better than the other three conditions combined.) To the extent that the elaborated-after condition can be singled out as superior to the other conditions that provided some source of elaboration, we would like to conclude that author-provided elaborations are superior to the ones that the reader can generate independently. We believe, however, that such a conclusion would be premature.

We hypothesized at the outset that author-provided elaborations would help in a skill-learning situation because, unlike simple declarative learning situations, good performance requires judging the appropriateness of the stored information to the task context and generalizing contexts of application (for a rule) to novel situations.⁴ Given that we have found that author-provided elaborations do help in a skill-learning situation, it becomes interesting to determine whether the advantage of elaborations is related to identifying contexts of application, as hypothesized, or due to some other aspect of a skill-learning situation that is not shared by the standard declarative tests.

We have implied that the skill-learning tests tap understanding in a way that the declarative tests of prior research do not. On the other hand, the difference between the two learning situations is not simply that one requires a deeper understanding or greater ability to recognize contexts of application. For example, acquiring the skill of using a computer requires that subjects learn the exact syntax of commands that will be needed to perform a task. Many other skills (e.g., mathematics and writing) involve abstract rules for manipulating symbols. Learners not only must remember an abstract rule, but also must produce a specific instantiation of the rule that is appropriate for the task at hand.⁵

Given that subjects who did not have advance knowledge of the tasks performed so much more efficiently after reading author-provided elaborations, we wondered whether the benefit of the elaborations was for choosing the most efficient applications of commands or for learning to implement the command syntax. For example, a number of elaborations in the manual gave advice about when to use shortcuts (such as wild-card characters) in commands. If these elaborations helped subjects remember to use the shortcuts at appropriate times, then these subjects should have been able to complete the tasks with fewer commands. On the other hand, the manual also contained many examples of syntactically correct commands and detailed explanations of the notations. If these elaborations helped subjects formulate correct commands, then the greater overall efficiency would be due not to

the efficiency of the solution, but rather to the subjects' abilities to carry it out.

One source of information on this question is the on-line record of the subjects' interactions with the computer. We analyzed 20 on-line protocols from the elaborated-after and unelaborated-after conditions, 10 for each version of the manual. The commands that the subjects issued were categorized into five types⁶: (1) productive moves—syntactically correct commands that carry out a target action or that enable one; (2) verification moves—commands that check whether a previous command had the desired effect; (3) execution errors—commands that contain one or more syntactic errors; (4) goal-specification errors—wrong command issued or failure to perform a prerequisite action, showing that subject may have some misconception about current state of the computer or the capabilities of a command; (5) recovery moves—commands to gain information after an error or to undo its effects.

Table 2 displays the distribution of commands that fall into each of these five categories for each subject as a function of type of manual studied. If the elaborations had helped subjects generate more efficient solution strategies, then we would have expected subjects in the elaborated group to need fewer productive moves; however, although the elaborated group did issue slightly fewer productive moves, the differences for both the productive moves and the verification moves are negligible. On the other hand, subjects who read elaborated manuals issued less than half as many commands that were syntactically incorrect, and this difference was significant [$t(18) = 2.6$, $MSE = 4.1$, $p < .05$]. Furthermore, they made half as many goal-specification errors and took only half as many moves to recover from an error. The latter contrast was also reliable [$t(18) = 2.3$, $MSE = 4.1$, $p < .05$]. Thus, subjects who studied the elaborated manual not only generated correct syntax more often, they were also better at knowing what commands to generate and at fixing commands that were wrong.

The results of the protocol analysis suggest that the benefits of elaborations may be due to the syntactic elaborations, those that specify how to perform a procedure. In Experiment 2, we separated the syntactic elaborations from the general conceptual elaborations, varying the presence of the two types orthogonally. Conceptual elaborations should be important in a skill task if the facilitating effect of elaborations for skill acquisition results from

Table 2
Mean Steps per Subject for Five Kinds of Actions as a Function of Version of Manual, Experiment 1

	Elaborated Manual	Unelaborated Manual
Productive Moves	27.7	33.7
Verification Moves	11.0	12.3
Execution Errors	9.5	20.2
Goal-Specification Errors	7.3	13.5
Recovery Moves	11.3	20.8

a need to determine the appropriateness of the stored information to the task constraints. On the other hand, if elaborations help in a skill situation because of the difficulty in learning how to generate syntactically correct commands, then only the syntactic elaborations or examples will facilitate performance.

EXPERIMENT 2

In Experiment 1, we used manuals that either elaborated both concepts and syntax or elaborated neither. For Experiment 2, we created two additional versions of the manuals, so that we could orthogonally vary the elaboration for concepts and command syntax.

In addition to determining whether both syntactic and conceptual information benefited equally from elaborations, we also wondered whether benefit from different types of elaborations varied as a function of subjects' experience. Specifically, would computer-experienced users benefit from the same types of elaborations as would computer novices when learning to use a new operating system? We expected these two groups to have almost complementary needs. Experienced users already understand the general concepts behind computer systems. They need to know how those concepts are instantiated in the new system and might be distracted and bored by elaborations on concepts they already understand and by long expositions on how the commands work. Conversely, novices lack a clear conception of what a computer operating system can do. They might benefit from a longer discussion of these new concepts. Both experienced users and novices, on the other hand, might benefit from seeing concrete examples of the command syntax. Although experienced users may be better able to parse the standard abstract syntax specifications found in most computer manuals, they, too, should prefer more informative specifications and instances from which to generalize the rule.

Method

Design. For this experiment, we used a $2 \times 2 \times 2$ between-subjects factorial design: computer experience (experienced vs. novice computer users), conceptual elaborations (present or absent from the manual), and elaborations of command syntax (present or absent from the manual). Subjects were screened for prior computer experience and, if appropriate, were assigned to one of the experience groups. Within each experience group, subjects were randomly assigned to one of the four manual conditions.

Subjects. Seventy-two subjects were paid \$6 each for participating. The 40 novices were students or staff members from C-MU, University of Pittsburgh, or Allegheny Community College who had taken at most one programming course and claimed some familiarity with an interactive computer operating system, but not one running UNIX, VMS, or DOS. The 32 experienced subjects were graduate students, faculty, or staff from C-MU or University of Pittsburgh who knew at least two programming languages and were familiar with either the UNIX or VMS operating systems, both of which support subdirectories. The experienced subjects also had experience using microcomputers, but not the IBM-PC.

Materials. The elaborations in the manuals varied along two dimensions. A manual was concept rich if it contained elaborations

on the purpose of the commands and when it was a good idea to use them. Concept-rich manuals also contained elaborations on basic topics such as disk drives, subdirectories, and paths. All of these elaborations were omitted from the concept-poor versions. A manual was syntax rich if it contained elaborations on how to issue a command, such as examples of correct commands and descriptively rich format statements. These elaborations were omitted from the syntax-poor versions. These variables were combined factorially to create four versions of the manuals. The syntax-rich elaborations are illustrated in Appendix C and the concept-rich elaborations in Appendix D.

Procedure. The procedure for this study was very similar to that of Experiment 1, except that no subjects were given advance information about the tasks before reading the manual. Subjects read their assigned manuals at their own paces (for a maximum of 1 h). Subjects were told that they should review the important points of the manual as often as time permitted, because the manual would not be available while they performed the task. After the reading period was over, the manuals were removed and subjects were asked to perform six tasks on the computer to the best of their ability. Since the experienced computer users might give ceiling performance on tasks that novices find challenging, and since novices might give floor performance on tasks that experienced subjects find challenging, half the tasks were designed to be relatively easy and half the tasks were designed to be relatively difficult. The instructions for the six tasks were as follows:

Task 1. Before you, in drive A, is a diskette containing a number of files. Some of these files have the word "PART" in their names, such as file "PART.1." We want you to change the names of these files. The new name that you should give each file appears on the first line of that file. So, to find the new name, you must inspect the contents of each file that now has "PART" in its name and then give the file the name that you find on the first line of that file.

Task 2. Next, you should create two directories on this diskette. One directory is to be named PROGRAMS and the other named DATA. Move all the files that have the word "Program" in their names into the PROGRAMS directory. And, similarly, move the "Data" files into the DATA directory. You do not want any Program or Data files to remain in the root directory of the diskette in drive A.

Task 3. The diskette in drive B contains a subdirectory named "NEW." We would like you to move all of the files in this subdirectory to the root directory of the diskette in drive A. You do not want any files to remain in the directory on drive B.

Task 4. The root directory on the diskette in drive A contains three files with .DCR extensions and three files with .PRG extensions. Each .DCR file matches one of the .PRG files in its stem name. That is, the SORT.DCR file corresponds to the SORT.PRG file, and so on. We would like you to create three new files by appending each .PRG file to the end of the corresponding .DCR file. The new files should all have the extension .CEE. So, for example, you will create a file called SORT.CEE which would contain the contents of SORT.DCR followed by the contents of SORT.PRG. You should only need to issue one command to accomplish this task.

Task 5. The root directory on the diskette in drive A contains a directory called EXP83. EXP83 contains another directory called SUBJ.FLS. We want you to create a file called EXP83.LST in the EXP83 directory. The contents of this file, EXP83.LST, should be a list of the names of the files in the SUBJ.FLS directory.

Task 6. We want you to eliminate the directory named SOURCE and everything it contains from the root directory of the diskette in drive A.

Tasks 1, 2, and 3 were considered relatively easy. Task 4 required subjects to compose a complex command with wild cards. Task 5 required using a variant of the COPY command to create

a pipe from the keyboard to a file. Task 6 required sophisticated path specifications in commands to operate on deeply embedded files and subdirectories.

Results

The scoring procedure was the same as that used for Experiment 1. The results described below reflect the scores of 72 subjects; 10 novice subjects and 8 experienced subjects in each of the four manual conditions. Experienced subjects took less time to read the manual than did novices [42 min vs. 55 min; $F(1,64) = 27.0$, $MSe = 113$, $p < .01$], and manuals that contained conceptual elaborations were read more slowly than those that did not [51 min vs. 46 min; $F(1,64) = 4.0$, $MSe = 113$, $p < .05$].

The classification of tasks as easier or more difficult was validated by the significantly better performance for easier tasks than for the more difficult ones. Subjects completed 80% of the easy tasks, compared to only 64% of the hard tasks [$F(1,64) = 27.0$, $MSe = .03$, $p < .01$]. This advantage for the easier tasks was also found for the other performance measures.

Table 3 presents the mean percentage of tasks successfully completed as a function of task difficulty, experience, and types of elaboration. As in Experiment 1, there were no significant differences in ability to complete the task as a function of type of elaboration. This supports the idea that even our minimal manuals were adequate for learning to perform the tasks. This null result is not caused by noisy data or insensitive measures. The measures were sensitive enough to reveal significant differences in completion due to difficulty of task (as reported earlier), and to reveal differences in successful performance as a function of prior experience [$F(1,64) = 24.25$, $MSe = .08$, $p < .01$]. There is a suggestion that conceptual elaborations interfered for novices, but this effect was not reliable.

Although there were main effects of experience on all measures of performance, experience did not interact with manipulations of manual content.⁸ Similarly, the variable of task difficulty produced significant results, but did not interact with any of the variables of interest. Therefore,

Table 3
Percentage of Tasks Successfully Completed as a
Function of Experience, Task Difficulty, and
Type of Elaborations, Experiment 2

	Syntax Rich		Syntax Poor	
	Concept Rich	Concept Poor	Concept Rich	Concept Poor
Easy Tasks				
Novices	.64	.73	.61	.71
Experienced Users	.95	1.00	.95	.91
Hard Tasks				
Novices	.47	.58	.41	.41
Experienced Users	.89	.87	.90	.72
Mean				
Novices	.55	.65	.54	.61
Experienced Users	.91	.95	.92	.84

Table 4
Performance on Task as a Function of Type of Elaboration-
Availability in the Manual, Experiment 2

	Syntax Rich		Syntax Poor	
	Concept Rich	Concept Poor	Concept Rich	Concept Poor
Mean Time on Task in Minutes (All Tasks)	37.4	37.7	43.5	45.9
Mean Number of Commands Issued (All Tasks)	71.7	73.7	88.7	92.4
Proportion of Commands Issued Per Minimum Step Required (Completed Tasks Only)	3.5	3.5	4.5	5.3

we have collapsed over experience and task difficulty in our description of the remaining results. Table 4 presents results from Experiment 2 using the same measures as those presented in Table 1 for Experiment 1. The data are presented as a function of whether or not the manual contained concept elaborations and whether or not it contained syntax elaborations.

The same pattern emerges in all these measures: facilitation when syntax elaborations are present, but no effect from conceptual elaborations. Subjects who saw the syntax elaborations were marginally faster than those who did not [$F(1,64) = 2.84$, $MSe = 319$, $p < .10$]. Subjects issued significantly fewer commands when the manual contained syntactic elaborations [$F(1,64) = 5.7$, $MSe = 1005$, $p < .05$]. As in Experiment 1, we computed the mean number of commands for a completed task as a proportion of the minimum number required. There is a significant effect of syntactic elaborations [$F(1,56) = 6.69$, $MSe = 4$, $p < .05$], but no effect of conceptual elaborations.⁹

GENERAL DISCUSSION

In both Experiments 1 and 2, elaborations improved the quality of skill performance. In Experiment 1, performance was least efficient when the manual did not contain any elaborations and the subjects did not know in advance what tasks they would be asked to perform. In Experiment 2, performance was again least efficient when the manual contained no elaborations. This was true regardless of prior computer experience and regardless of the difficulty of the required tasks.

Two versions of the manuals used in Experiment 2 were essentially identical to those used in Experiment 1. The version without any elaboration contained few changes from the original unelaborated version. Similarly, the manual that contained both syntax and concept elaborations was essentially the same as the elaborated manual in Experiment 1. By creating two new versions, one with only syntax elaborations and one with only concept elaborations, we were able to determine whether both types of elaborations contributed equally to the new result that

author-provided elaborations help in a skill-learning situation. We found that only the syntactic elaborations systematically improved performance.

We began this paper by suggesting that elaborations might help in a skill-acquisition situation, even though they do not seem to facilitate performance in a declarative knowledge test. The reason, we argued, was that conceptual elaborations might help subjects determine which procedure to use in a novel situation. We also expected syntactic elaborations and examples to help subjects decide how to issue specific commands to the computer. The data support the idea that syntactic elaborations help people determine exactly how to implement a procedure. The useful elaborations were those that explained the syntactic notation and illustrated how syntactically correct commands should look: without these elaborations, subjects had to rely exclusively on abstract syntactic rules.

Despite the failure to show facilitation from concept elaborations, it seems that the benefit of elaborations goes beyond helping subjects use syntax correctly in issuing commands. The pattern of errors and specific use of commands from the analysis of the on-line protocols in Experiment 1 suggests that subjects got more out of the text generally when the documentation was elaborated (see Table 2).

An important question that remains is why the syntax examples helped performance to the exclusion of the concept elaborations. Perhaps the syntax examples more closely matched what the subject needed to do. The concept elaborations were not written with any specific task in mind. Although the elaborations on command syntax were also not written with a particular task in mind, many of the commands described in the manual were tested by the tasks. Conceivably, if the mapping between the concept examples and the task requirements were as close as the map between the syntax examples and the task requirements, performance would also benefit from the conceptual elaborations.

Consistent with this view, Ross (1984) has shown that performance in a new skill domain is influenced by prior examples when these examples match on superficial features. It appears that "reminders" to prior examples depend more on the closeness of superficial features of the example than on the appropriateness of the example. Pirolli and J. R. Anderson (1985) report a study on learning LISP that indicates that it is not the example per se that helps performance, but whether the example is used to illustrate how to do the task rather than to simply clarify what happens when a procedure is used.

The Procedural/Declarative Knowledge Distinction Revisited

Given the result that only syntactic elaborations facilitated performance, the question naturally arises as to whether elaborations have a different status for learning a skill than for learning facts, or whether syntactic elaborations help performance, regardless of the nature of the task, whereas other types of elaborations do not. In order

to answer this question, we conducted a replication of Experiment 2, substituting a declarative knowledge test for the skill-learning test.

The declarative knowledge test consisted of 24 true-false statements, such as "The RMDIR command can be used to delete any directory, including the root directory" (False) and "When typing the location information for the parameters in your commands, do not leave spaces between the drive and path specifications" (True). The questions were written with the intention that answers would not require subjects to recognize contexts of application of a fact, but rather would depend on the subject's ability to retrieve studied facts from memory. To reduce the possibility that prior knowledge would affect performance on this test, given that it only required a binary decision, we pretested the questions with subjects who had not read the manuals. We asked 43 people to answer the true-false questions and discarded any question whose accuracy was below 25% or above 75% correct.

In the experiment, 32 subjects were run in groups of 10-12, and were randomly assigned to one of the four instructional conditions. All subjects were inexperienced computer users who had taken at most one programming course. They were allowed to read the manual at their own paces for up to 1 h. After they finished studying the manual, they were given the true-false test on a sheet of paper. They were given unlimited time to complete the test, but most finished in about 10 min. The entire experiment lasted about 75 min (with subjects averaging about 50 min to read the document and 25 min to read instructions and administrative material).

The results of the true-false test are presented in Table 5 as a function of the extent of syntactic and conceptual elaboration. The differences among conditions are quite small, and there are no main effects or interactions. It seems that the documentation does not produce the same effect when the test taps declarative rather than procedural knowledge. That is, subjects who studied manuals containing syntactic elaborations did not out-perform subjects in the other conditions. On the other hand, the results do not show the same pattern as that found by Reder and J. R. Anderson (1980, 1982) when they looked at fact learning as a function of extent of elaboration. That is, subjects who saw the unelaborated manual did not perform better than subjects who read the longer, elaborated manuals.

A reasonable conclusion from this pattern of results is that the advantage of elaborations found in Experiments 1 and 2 was not due to the performance measure per se; that is, the conjecture made earlier that elaborations help

Table 5
Mean Percent Correct on True/False Questions that Tested Materials Used, Experiment 2

	Concept Rich	Concept Poor
Syntax Rich	.68	.63
Syntax Poor	.65	.61

procedural knowledge, but not declarative knowledge, has not received much support. A more likely conclusion is that the benefit or liability of elaborations is a function of their quality and relevance to the main points that they support.

In summary, this research shows that the findings of Reder and J. R. Anderson (1980, 1982) do not extend to the domain of skill acquisition; however, the advantage of elaborations seems to be restricted to providing the reader with explanations and concrete illustrations of how the skill is performed. At this time, there is little evidence that other types of author-provided elaborations are beneficial to skill learning. On the other hand, this conclusion may be affected by the nature of the commands being taught and the tasks being tested.¹⁰ Appropriate command use requires that the user know when to use the command and how to use the command. The latter requirement was facilitated by syntactic elaborations, whereas the conceptual elaborations did not help the former. It may have been that the applicability issues were trivial in this task and could not benefit from conceptual elaborations. Alternatively, if the syntax of the commands had been more familiar, the syntactic elaborations might not have been as facilitating.

REFERENCES

- ALLWOOD, C. M., WIKSTROM, T., & REDER, L. M. (1982). Effects of presentation format on reading retention: Superiority of summaries in free recall. *Poetics*, 11, 145-153.
- ANDERSON, J. R. (1974). Retrieval of propositional information from long-term memory. *Cognitive Psychology*, 5, 451-474.
- ANDERSON, J. R. (1980). *Cognitive psychology and its implications*. San Francisco: Freeman.
- ANDERSON, J. R., & REDER, L. M. (1979). An elaborative processing explanation of depth of processing. In L. S. Cermak & F. I. M. Craik (Eds.), *Levels of processing in human memory*. Hillsdale, NJ: Erlbaum.
- ANDERSON, R. C., & BIDDLE, W. B. (1975). On asking people questions about what they are reading. In G. H. Bower (Ed.), *The psychology of learning and motivation* (Vol. 9). New York: Academic Press.
- ANDERSON, R. C., & PICHERT, J. W. (1977, April). *Recall of previously unrecallable information following a shift in perspective* (Tech. Rep. No. 41). Urbana: University of Illinois, Center for the Study of Reading. (ERIC Document Reproduction Service No. ED 142 974)
- ARKES, H. R., & FREEDMAN, M. R. (1984). A demonstration of the costs and benefits of expertise in recognition memory. *Memory & Cognition*, 12, 84-89.
- BARNARD, P. J., HAMMOND, N. V., MORTON, J., LONG, J. B., & CLARK, I. A. (1981). Consistency and compatibility in human/computer dialogue. *International Journal of Man-Machine Studies*, 15, 87-134.
- BARTLETT, F. C. (1932). *Remembering: A study in experimental and social psychology*. Cambridge, England: Cambridge University Press.
- BOBROW, S., & BOWER, G. H. (1969). Comprehension and recall of sentences. *Journal of Experimental Psychology*, 80, 455-461.
- BOWER, G. H. (1976, August). *Comprehending and recalling stories*. Section 3, Presidential Address presented at the meetings of the American Psychological Association, Washington, DC.
- BOWER, G. H., BLACK, J. B., & TURNER, T. J. (1979). Scripts in memory for text. *Cognitive Psychology*, 11, 177-220.
- BRADSHAW, G., & ANDERSON, J. R. (1982). Elaborative encoding as an explanation of levels of processing. *Journal of Verbal Learning & Verbal Behavior*, 21, 165-174.
- BRANSFORD, J. D. (1979). *Human cognition: Learning, understanding and remembering*. Belmont, CA: Wadsworth.
- BROWN, A. L., SMILEY, S. S., DAY, J. D., TOWNSEND, M. A. R., & LAWTON, S. C. (1977). Intrusion of a thematic idea in children's comprehension and retention of stories. *Child Development*, 48, 1454-1466.
- BUGELSKI, B. R. (1962). Presentation time, total time, and mediation in paired associate learning. *Journal of Experimental Psychology*, 63, 409-412.
- CHIESI, H. L., SPILICH, G. J., & VOSS, J. F. (1979). Acquisition of domain-related information in relation to high and low domain knowledge. *Journal of Verbal Learning & Verbal Behavior*, 18, 257-273.
- COOPER, E. H., & PANTLE, A. J. (1967). The total-time hypothesis in verbal learning. *Psychological Bulletin*, 68, 221-234.
- CRAIK, F. I. M., & TULVING, E. (1975). Depth of processing and the retention of words in episodic memory. *Journal of Experimental Psychology: General*, 104, 268-294.
- DOOLING, D. J., & CRISTIAANSEN, R. E. (1977). Episodic and semantic aspects of memory for prose. *Journal of Experimental Psychology: Human Learning & Memory*, 3, 428-436.
- MICROSOFT, INC. (1983). *IBM Personal Computer Language Series. Disk Operating System, Version 2.00*. Boca Raton, FL: IBM.
- MANDL, H., & BALLSTAEDT, S. (1981, September). *Effects of elaboration on recall of texts*. Paper presented at the International Symposium on Text Processing, Fribourg, Switzerland.
- MANDL, H., SCHNOTZ, W., & TERGAN, S. (1984, April). *On the function of examples in instructional texts*. Paper presented at the AERA meeting, New Orleans.
- OWENS, J. E., & BOWER, G. H. (1977, August). *Character point of view in text comprehension and memory*. Paper presented at the meeting of the American Psychological Association, San Francisco.
- PEPPER, J. (1981). Following students' suggestions for rewriting computer programming textbooks. *American Education Research Journal*, 18, 259-269.
- PHIFER, S., McNICKLE, B., RONNING, R., & GLOVER, J. (1983). *The role of details in the recall of major ideas in text*. Unpublished manuscript.
- PIROLLI, P. L., & ANDERSON, J. R. (1985). The role of learning from examples in the acquisition of recursive programming skills. *Canadian Journal of Psychology*, 39, 240-272.
- POSNER, M. I. (1973). Abstraction and the process of recognition. In G. H. Bower (Ed.), *The psychology of learning and motivation*, III. New York: Academic Press.
- POSTMAN, L. (1971). Organizing and interference. *Psychological Review*, 78, 290-302.
- POSTMAN, L., & STARK, K. (1969). Role of response availability in transfer and interference. *Journal of Experimental Psychology*, 79, 168-177.
- REDER, L. M. (1976). The role of elaborations in the processing of prose. *Dissertation Abstracts International*, 37, 5405B. (University Microfilms No. 77-8016)
- REDER, L. M. (1979). The role of elaborations in memory for prose. *Cognitive Psychology*, 11, 221-234.
- REDER, L. M. (1982). Elaborations: When do they help and when do they hurt? *Text*, 2, 211-224.
- REDER, L. M. (1985). Techniques available to author, teacher and reader to improve retention of main ideas of a chapter. In J. Segal, S. Chipman, & R. Glazer (Eds.), *Thinking and learning skills: Current research and open questions* (Vol. 2). Hillsdale, NJ: Erlbaum.
- REDER, L. M., & ANDERSON, J. R. (1980). A comparison of texts and their summaries: Memorial consequences. *Journal of Verbal Learning & Verbal Behavior*, 19, 121-134.
- REDER, L. M., & ANDERSON, J. R. (1982). Effects of spacing and embellishment on memory for the main points of a text. *Memory & Cognition*, 10, 97-102.
- REDER, L. M., CHARNEY, D., & MORGAN, K. (1984, August). *The role of elaborations in learning a skill from an instructional text* (Tech. Rep. No. 1). Pittsburgh: Carnegie-Mellon University.
- RICKARDS, J. P. (1979). Adjunct postquestions in text: A critical review of methods and processes. *Review of Educational Research*, 49, 181-186.
- ROHWER, W. D., JR., & AMMON, M. S. (1971). Elaboration training and paired-associate learning efficiency in children. *Journal of Experimental Psychology*, 62, 376-383.

- ROHWER, W. D., JR., LYNCH, S., LEVIN, J., & SUZUKI, N. (1967). Pictorial and verbal factors in the efficient learning of paired-associates. *Journal of Experimental Psychology*, 58, 278-284.
- ROHWER, W. D., JR., LYNCH, S., SUZUKI, N., & LEVIN, J. (1967). Verbal and pictorial facilitation of paired-associate learning. *Journal of Experimental Child Psychology*, 5, 294-302.
- ROSS, B. (1984). Reminders and their effects in learning a cognitive skill. *Cognitive Psychology*, 16, 371-416.
- ROTHKOPF, E. Z., & BILLINGTON, M. J. (1983). Passage length and recall with test size held constant: Effects of modality, pacing, and learning set. *Journal of Verbal Learning & Verbal Behavior*, 22, 667-681.
- SCHALLERT, D. L. (1976). Improving memory for prose: The relationship between depth of processing and context. *Journal of Verbal Learning & Verbal Behavior*, 15, 621-632.
- SHIFFRIN, R. M., & SCHNEIDER, W. (1977). Controlled and automatic human information processing: I. Detection, search, and attention. *Psychological Review*, 84, 1-66.
- STEIN, B. S., & BRANSFORD, J. D. (1979). Constraints on effective elaborations: Effect of precision and subject generation. *Journal of Verbal Learning & Verbal Behavior*, 18, 769-778.
- SULIN, R. A., & DOOLING, D. J. (1974). Intrusion of a thematic idea in retention of prose. *Journal of Experimental Psychology*, 103, 255-262.
- WEINSTEIN, C. E. (1978). Elaboration skills as a learning strategy. In H. F. O'Neil, Jr. (Ed.), *Learning strategies*. New York: Academic Press.

NOTES

1. Dooling and Cristiaansen (1977), for example, explored to what extent the false alarms were due simply to response bias as opposed to differential encoding. Although part of the result is due to response bias, part is clearly due to encoding differences.

2. Glover and his colleagues (Phifer, McNickle, Ronning, & Glover, 1983) found that the summaries produced worse performance than did the elaborated texts, if subjects were given a small fixed amount of time

per sentence. The sentence-presentation rate for a subject was the average rate at which he or she read sentences from a novel. Under these circumstances, the subjects could not comprehend the summaries, and their performances suffered. As described above, Reder and J. R. Anderson (1982) also equated reading time per sentence, but used a much slower presentation rate; under these conditions, performance was much better without elaborations.

3. Essay exams do not fall into the category of declarative tests as we define them. Writing an essay clearly calls for a deep understanding of a body of information and for selecting appropriate items from the relevant facts.

4. We wish to distinguish here between the type of skill required to use an IBM-PC and the type of skill required to follow a set of instructions for assembling a device or a piece of machinery. The latter does not require recognizing contexts of application.

5. The syntax in DOS is fairly intransigent. While there has been some attempt recently to develop operating systems and programming languages that tolerate misspellings, substitution of synonyms, variable order of arguments, and so forth (e.g., Barnard, Hammond, Morton, Long, & Clark, 1981), learning to use highly rigid syntactic rules is still an extremely common requirement.

6. A second independent judge coded a random sample of the protocols. The agreement between the two judges was quite high ($r = .90$); any disagreements were resolved to mutual satisfaction.

7. The word counts in the four versions were as follows: both concept-rich and syntax-rich elaborations, 10,686 words; only concept-rich elaborations, 8,366 words; only syntax-rich elaborations, 5,699 words; no elaborations, 3,428 words.

8. It is unlikely that the lack of an interesting result due to expertise is due to our novices' having had some computer experience. We found that novices who had no background whatsoever could not even begin to perform the tasks of the experiment, causing a floor effect among the conditions. We selected subjects for the novice condition whom we felt were as inexperienced as possible.

9. The degrees of freedom differ for the completed tasks measures because some subjects did not successfully complete any tasks.

10. We would like to thank an anonymous reviewer for making this point.

APPENDIX A The Commands Taught in the User's Manual

COMMAND	FUNCTION
DIR	List the files in a directory
MKDIR	Create (or "make") a subdirectory
CHDIR	Change default directory assignment
RMDIR	Eliminate (or "remove") a subdirectory
TYPE	Display the contents of a file
RENAME	Change the name of a file
COPY	Create a duplicate copy of a file Combine or append files together Transfer data between system devices
ERASE	Eliminate a file

^Numlock	Freeze the screen
^Break	Abort the current command
d	Change the default drive assignment

APPENDIX B
 Samples of Four Types of Elaboration from the Manual Used in Experiment 1

Meta-statement

ELABORATED

Since the computer has two disk drives which can each contain a diskette, you must specify whether the file you want is in drive A or drive B when you give the computer a command. If your command doesn't specify which drive contains the file, the computer automatically assumes that it can find the file in the "default" drive. The next section explains what the "default drive" is, and how to tell the computer to look on a different drive if necessary.

UNELABORATED

Since the computer has two disk drives which can each contain a diskette, you must specify whether the file you want is in drive A or drive B when you give the computer a command. If your command doesn't specify which drive contains the file, the computer automatically assumes that it can find the file in the "default" drive.

Definition

ELABORATED

The B: ("B-colon") in the command stands for the right-hand disk drive. The colon signals the computer that the letter or word preceding it is a "device" rather than the name of a command or file. Devices are pieces of computer hardware, such as disk drives, a printer or even the keyboard. After you enter the command, the B> prompt will appear on the screen. From now on the computer will automatically look for files on drive B.

UNELABORATED

The B: in the command stands for the right hand disk drive. From now on, the B> prompt appears on the screen and the computer will automatically look for files on drive B.

Analogy

ELABORATED

When you give the computer a command concerning a file, such as TYPE, ERASE or COPY, the computer looks for the file on a "diskette." A diskette, also known as a "floppy disk," is similar to a small, flexible phonograph record, except that instead of storing sounds, it contains information which the computer can read, add to or delete. All the files you create on the computer are stored on diskettes. So, in order to work on your files, you must insert the diskette that contains them into the computer. You insert a diskette into one of the two "disk drives" on the front of the computer cabinet. The drive on the left is called drive A, and the one on the right is drive B.

UNELABORATED

When you give the computer a command concerning a file, such as TYPE, ERASE or COPY, the computer looks for the file on a "diskette." To use a diskette, you insert it into one of the two "disk drives" on the front of the computer cabinet. The drive on the left is called drive A, and the one on the right is drive B.

Example

The elaborated and unelaborated versions are for the most part identical except for the addition of the example (italicized here)

Using COPY to Combine Files

You can use COPY to combine files, appending a copy of one file to the end of another file.

APPENDIX B (continued)

FORMAT

The format of the command is:

```
COPY [Loc & name first file - next file - ] [Loc & name combined file]
```

[Loc & name first file - next file] refers to a list of the files you want to "add" together. The names of the files are typed with plus (+) signs between them. You need to specify location information for each filename in the list in the usual way, with drive and path specifications. When several filenames are listed in this manner the COPY command results in a new file in which the contents of the first file on the list appear first, followed by the contents of the second file, then the contents of the third file and so on. So be sure that the files in the list appear in the order in which you want them combined.

[Combined file] refers to the new file that will contain the combined files, what you want to call this file, and where in the directory structure you want it to go. Specify the location in terms of a drive and a path to a directory as usual. Type the name you would like to give the file at the end of the path.

For example, suppose you write a report in sections, with each section in a separate file. You want to format and print the report as one file, so you combine the sections into one file. The following command takes three files, *INTRO.MSS*, *BODY.MSS*, and *CONCL.MSS* and combines them into a new file called *REPORT*.

```
A> COPY B:INTRO.MSS+B:BODY.MSS+B:CONCL.MSS REPORT <ENTER>
```

The combined file, *REPORT* will consist of the introduction, the body, and the conclusion.

APPENDIX C

Excerpt of Manual from Experiment 2 Illustrating RICH SYNTAX Elaborations

CHANGING THE CURRENT DIRECTORY -- CHDIR

The CHDIR command allows you to designate a directory as the "current" directory for a drive, so that the computer will automatically look there for files or subdirectories mentioned in your commands. You can designate a current directory for each disk drive independently.

FORMAT

```
CHDIR [loc and name of new current directory]
```

You can use the abbreviation CD in the command instead of typing CHDIR.

[Location of new current directory] refers to the path to the directory you want to designate as the new current directory. The last directory name on the list should be the name of the directory you want to designate.

For example, the command below designates a subdirectory called PASCAL as the new current directory in drive B.

```
A> CHDIR B:\PROGRAMS\PASCAL <ENTER>
```

The first symbol in the path is a backslash (\). This means that the path to the new current directory starts with the root directory of the diskette in drive B. The path indicates that the root directory contains a subdirectory called PROGRAMS, and that PROGRAMS contains PASCAL, the directory you want to designate as the "new" current directory. As usual, the amount of location information you need to provide depends on which directory was last designated as the current directory for the drive.

To change the current directory back to the root directory, give a command like the following.

```
A> CHDIR B:\ <ENTER>
```

The backslash (\) in the commands above symbolize the root directory. So the command above changes the current directory for drive B to the root directory.

If you forget which directory is the current directory, the computer can remind you. Enter a CHDIR command without specifying a location. The computer will display the path from the root directory to the current directory or a backslash if you are still in the root directory.

APPENDIX D
Excerpt of Manual from Experiment 2 Illustrating
RICH CONCEPT Elaborations

CHANGING THE CURRENT DIRECTORY -- CHDIR

The CHDIR command (short for "change directory") allows you to designate a directory as the "current" directory for a drive so that the computer will automatically look there for files or subdirectories mentioned in your commands. You can designate a current directory for each disk drive independently. Changing the current directory on the diskette in drive A does not affect the current directory on drive B.

The root directory is automatically designated as the current directory for each drive when you first start up the computer. It is useful to designate a subdirectory as the current directory when you will be working primarily on the files in that subdirectory. Then you won't have to specify the path to the subdirectory in each command you issue.

FORMAT

The format of the command is:

CHDIR [[d:]path]

You can use the abbreviation CD in the command instead of typing CHDIR.

If you designate a subdirectory as the new current directory, the computer will carry out all the subsequent commands within that directory, unless you specify a path to another directory. To change the current directory back to the root directory, use a backslash as the path.

If you forget which directory is the current directory, the computer can remind you. Enter a CHDIR command without specifying a location. The computer will display the path from the root directory to the current directory, or "I", if you are still in the root directory.

(Manuscript received February 25, 1985;
revision accepted for publication July 8, 1985.)