JOURNAL OF MEMORY AND LANGUAGE 25, 507–524 (1986)

The Acquisition of Procedures from Text: A Production-System Analysis of Transfer of Training

DAVID E. KIERAS AND SUSAN BOVAIR

University of Michigan

Learning a cognitive skill from written instructions can be viewed as consisting of converting the propositional content of the written material into a representation of procedural knowledge, such as production rules. In a transfer of training experiment, subjects learned from step-by-step instructions a series of related procedures, in different training orders, for operating a simple device. The strong between-procedure transfer effects were predicted by a simple model of transfer in which individual production rules can be transferred or re-used in the representation of a new procedure if they had been used in a previously learned procedure. Apparently, this transfer mechanism acts on declarative propositional representations of the production rules, suggesting that it is more similar to comprehension processes than to conventional practice mechanisms, or to Anderson's learning principles (1982, *Psychological Review*, 89, 369–406; 1983, *The architecture of cognition*, Cambridge, MA, Harvard Univ. Press). © 1986 Academic Press, Inc.

Quite often people must learn procedures from written instructions, *procedural text*, which describe procedures explicitly. In the context of the currently developing theory of procedural knowledge and cognitive skill (Anderson, 1982, 1983), this task must involve the formation of production rules from the declarative content of the text. While Anderson has characterized how production rules can be formed once a suitable declarative representation is present, relatively little is known about the construction of the declarative representation itself, even when the input is procedural text. The results reported here provide an initial characterization of this process.

Two general conclusions will be presented: First, a production rule representation can provide a very precise characterization of the relative difficulty of learning a set of related procedures. Second, apparently there are powerful comprehension processes that operate very early in learning a procedure from text, which construct and combine declarative representations of production rules. Thus, many of the important processes involved in learning a procedure from a text can take place before a procedural representation has been formed.

In the experiment reported in this paper, subjects learned procedures for operating a simple piece of equipment by reading step-by-step instructions. By measuring the reading time on individual steps, and the accuracy of execution of the procedure, it is possible to track the acquisition of individual production rules. Since the different procedures each subject learned were related to each other, some transfer of training was possible between procedures. The key result is tha. the amount of transfer is predicted very well from the similarities between the production system representations for the procedures. The reading times show that subjects can exploit this similarity immediately, while they are reading the instructions, suggesting that

This work was supported by the Office of Naval Research under Contracts N00014-84-K-0731, NR 667-473, and N00014-85-K-0138, NR 667-543. Requests for reprints should be sent to David Kieras, Technical Communications Program, College of Engineering, University of Michigan, Ann Arbor, MI 48109.

0749-596X/86 $3.00

language comprehension processes can be central in the acquisition and transfer of procedural knowledge.

Description of the Task

The subjects learned series of procedures for how to operate a device consisting of a simple control panel. The device used in this experiment was the same as that used in Kieras and Bovair (1984), in which the major manipulation was whether subjects were taught a mental model for the internal organization and structure of the device. But here, subjects learned the device by rote.

The device is a slope-front box with a simple front panel, shown in Fig. 1, consisting of four controls and four indicator lights. A laboratory computer detects the positions of the controls and turns the indicator lights on and off. The four controls consist of a toggle switch (SP), a three-position selector (ESS), and two push-buttons (FM and FS). The four indicator lights are labeled SPI, EBI, MAI, and PFI. The labels are based on the mental model used in Kieras and Bovair (1984). The goal of operating the device is to get the PFI indicator light to flash. Each procedure learned by the subjects consists of several steps, as illustrated in the step-by-step instructions in Tables 1 and 2. Table 1 is the procedure for a "normal" situation, in which the device is operating properly. Table 2 is the procedure for a "malfunction" situation, in which some fictitious internal component of the device is not operating. Depending on the nature of the malfunction, the device either can be made to work by an alternate procedure, or can not. The final step in each procedure is to signal success or failure in getting the device to work.

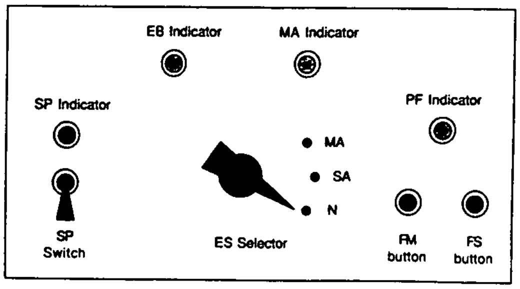

FIG. 1. The control panel device.

The behavior of the device can be most comprehensively described in terms of the mental model used in Kieras and Bovair (1984), but for brevity this will not be represented here. For purposes of this paper, the important property of the device is that since the procedures were based on a simple and consistent fictitious internal structure for the device, the procedures used to operate the device have a certain consistency and reasonableness, rather than being completely arbitrary. More detail on the behavior of the device can be found in Kieras and Bovair (1985).

The Operating Procedures

In the experiment, the subject was commanded to do either the *MA procedure* or the *SA procedure,* where these commands referred to which of the two ESS settings was to be tried first. A *malfunction situa-*

TABLE 1
EXAMPLE OF A NORMAL PROCEDURE

If the command is to do the MA procedure, then do the following:
Step 1. Turn the SP switch to ON.
Step 2. Set the ES selector to MA.
Step 3. Press the FM button, and then release it.
Step 4. If the PF indicator flashes, then notice that the operation is successful.
Step 5. When the PF indicator stops flashing, set the ES selector to N.
Step 6. Turn the SP switch to OFF.
Step 7. If the operation was successful, then type "S" for success.
Step 8. Procedure is finished.

TABLE 2
EXAMPLE OF A MALFUNCTION PROCEDURE

If the command is to do the MA procedure, then do the following:
Step 1. Turn the SP switch to ON.
Step 2. Set the ES selector to MA.
Step 3. Press the FM button, and then release it.
Step 4. If the PF indicator does not flash, then notice that there is a malfunction.
Step 5. If the EB indicator is on, and the MA indicator is off, then notice that the malfunction might be compensated for.
Step 6. Set the ES selector to SA.
Step 7. Press the FS button, and then release it.
Step 8. If the PF indicator does not flash, then notice that the malfunction can not be compensated for.
Step 9. Set the ES selector to N.
Step 10. Turn the SP switch to OFF.
Step 11. If the malfunction could not be compensated for, then type "N" for not compensated.
Step 12. Procedure is finished.

tion was defined as a situation in which first settings specified by the command would not work, and the procedure called for then trying the other settings. A total of 10 procedures were used, 2 *normal* and 8 *malfunction* procedures. The procedure steps are listed in Table 3. Tables 1 and 2 give examples of the step-by-step instructions for a normal and a malfunction procedure. Each procedure is labeled by the combination of the command (MA or SA) and the malfunction status, which is based on the fictitious internal components (NORMAL, XEB, XPB, XMA, XSA, XMA-XSA).

The eight malfunction procedures can be divided into two types. The first is those in which the alternate ESS setting might work, depending on the malfunction state. These were termed *possibly compensat-*

TABLE 3
PROCEDURES USED TO OPERATE CONTROL PANEL DEVICE

		MA procedures		
MA-NORMAL	MA-XEB	MA-XPB	MA-XMA	MA-XMA-XSA
(*1*) SP on	(*1*) SP on	(*1*) SP on	(*1*) SP on	(*1*) SP on
(*2*) ESS-MA	(*2*) ESS-MA	(*2*) ESS-MA	(*2*) ESS-MA	(*2*) ESS-MA
(*3*) FM push	(*3*) FM push	(*3*) FM push	(*3*) FM push	(*3*) FM push
(*4*) ESS-N	(*4*) ESS-N	(*4*) ESS-N	(*4*) ESS-SA	(*4*) ESS-SA
(*5*) SP off	(*5*) SP off	(*5*) SP off	(*5*) FS push	(*5*) FS push
(*6*) Tap "S"	(*6*) Tap "N"	(*6*) Tap "N"	(*6*) ESS-N	(*6*) ESS-N
			(*7*) SP off	(*7*) SP off
			(*8*) Tap "S"	(8) Tap "N"
		SA procedures		
SA-NORMAL	SA-XEB	SA-XMA-XSA	SA-XSA	SA-XPB
(*1*) SP on	(*1*) SP on	(*1*) SP on	(*1*) SP on	(*1*) SP on
(*2*) ESS-SA	(*2*) ESS-SA	(*2*) ESS-SA	(*2*) ESS-SA	(*2*) ESS-SA
(*3*) FS push	(*3*) FS push	(*3*) FS push	(*3*) FS push	(*3*) FS push
(*4*) ESS-N	(*4*) ESS-N	(*4*) ESS-N	(*4*) ESS-MA	(*4*) ESS-MA
(*5*) SP off	(*5*) SP off	(*5*) SP off	(*5*) FM push	(*5*) FM push
(*6*) Tap "S"	(*6*) Tap "N"	(*6*) Tap "N"	(*6*) ESS-N	(*6*) ESS-N
			(*7*) SP off	(*7*) SP off
			(*8*) Tap "S"	(*8*) Tap "N"

able malfunctions. In the second type, the alternate setting will not work, and so need not be tried. These were termed *noncompensatable malfunctions*. For example, the XEB state is a noncompensatable malfunction for either the MA or the SA command, and the MA-XMA and SA-XSA states are possibly compensatable malfunctions. This distinction was presented to the subjects as part of the overall instructions, in order to rationalize the details of the procedures.

THEORETICAL ANALYSIS

Transfer Effects

In earlier work with this device (see Kieras & Bovair, 1983) it was noticed that the time required to learn the procedures under rote conditions varied over a very wide range. The observed *training time* profile for the rote-learning subjects from this earlier work is shown in Fig. 2, which shows the training time for each procedure in the order that they were learned. Note that this order was fixed, rather than randomized, as would traditionally be done. Note that rather than being a smooth descending learning curve, there are large peaks for the times of the third, fifth, and ninth procedures. The number of steps in each procedure does not explain this pattern, because while the number of steps does vary for different procedures, the difference is not very large, and is frequently in the wrong direction. For example, procedure 5 has eight steps, which is more than procedures 3 and 4 with seven each, but procedure 6 has nine steps. Rather, the pattern could be explained by the observation that the first procedure appears to contain all new information, the second (the other normal procedure) contains only a little new information, the third (the first malfunction procedure) contains some new information, the fourth (the second malfunction procedure) very little, and the fifth (the first possibly compensatable malfunction) quite a lot. This intuitive result can be made well defined by transforming the instructions into production rules, which provide a precise characterization of what is to be learned in each procedure. By considering which of the rules were learned in a previous procedure, a quantitative measure of the amount of new information could be determined, namely, the number of new production rules that must be learned in each procedure.

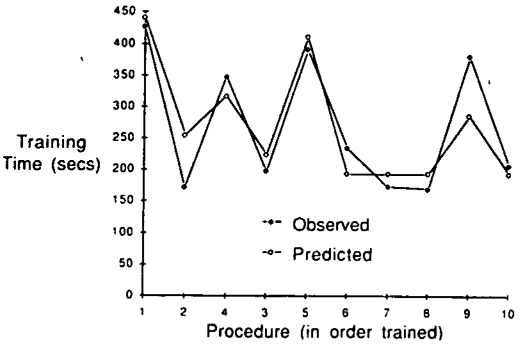

FIG. 2. Predicted and observed training times for the rote learning condition from Kieras and Bovair (1983).

Production Rule Representation

Table 4 provides an example production rule set for the procedure in Table 1. The syntax of these rules is very simple. Each rule consists of a name, a condition (following the IF), and a action (following the THEN). The condition is made up of a list of *clauses* which test for GOALS or NOTES in working memory, and are joined by AND, meaning that all of the clauses must be satisfied before the rule is fired. The actions can add or delete clauses from working memory, or operate on the device. The sequence of clauses added and deleted from working memory causes the rules to fire in the proper order to operate the device controls in the correct sequence. For example, the second rule in Table 4 is named MA-N-SP-ON, and can be paraphrased as follows: if a goal is to do the MA procedure and another goal is to do the step called SP-ON, then operate the SP control to the ON setting, wait for the device to respond, delete the goal to do this

TABLE 4
Example of Production Rules

```
(MA-N-START
IF (AND (TEST-GOAL DO MA PROCEDURE)
        (NOT (TEST-GOAL DO ??? STEP)))
THEN ((ADD-GOAL DO SP-ON STEP)))
(MA-N-SP-ON
IF (AND (TEST-GOAL DO MA PROCEDURE)
        (TEST-GOAL DO SP-ON STEP))
THEN ((OPERATE-CONTROL *SP ON)
      (WAIT-FOR-DEVICE)
      (DELETE-GOAL DO SP-ON STEP)
      (ADD-GOAL DO ES-SELECT STEP)))
(MA-N-ES-SELECT
IF (AND (TEST-GOAL DO MA PROCEDURE)
        (TEST-GOAL DO ES-SELECT STEP))
THEN ((OPERATE-CONTROL *ESS MA)
      (WAIT-FOR-DEVICE)
      (DELETE-GOAL DO ES-SELECT STEP)
      (ADD-GOAL DO FM-PUSH STEP)))
(MA-N-FM-PUSH
IF (AND (TEST-GOAL DO MA PROCEDURE)
        (TEST-GOAL DO FM-PUSH STEP))
THEN ((OPERATE-CONTROL *FM PUSH)
      (WAIT-FOR-DEVICE)
      (OPERATE-CONTROL *FM RELEASED)
      (DELETE-GOAL DO FM-PUSH STEP)
      (ADD-GOAL DO PFI-CHECK STEP)))
(MA-N-PFI-CHECK
IF (AND (TEST-GOAL DO MA PROCEDURE)
        (TEST-GOAL DO PFI-CHECK STEP)
        (LOOK *PFI FLASHING))
THEN ((ADD-NOTE OPERATION SUCCESSFUL)
      (DELETE-GOAL DO PFI-CHECK STEP)
      (ADD-GOAL DO ES-N STEP)))
(MA-N-ES-N
IF (AND (TEST-GOAL DO MA PROCEDURE)
        (TEST-GOAL DO ES-N STEP)
        (LOOK *PFI OFF))
THEN ((OPERATE-CONTROL *ESS N)
      (WAIT-FOR-DEVICE)
      (DELETE-GOAL DO ES-N STEP)
      (ADD-GOAL DO SP-OFF STEP)))
(MA-N-SP-OFF
IF (AND (TEST-GOAL DO MA PROCEDURE)
        (TEST-GOAL DO SP-OFF STEP))
THEN ((OPERATE-CONTROL *SP OFF)
      (WAIT-FOR-DEVICE)
      (DELETE-GOAL DO SP-OFF STEP)
      (ADD-GOAL DO TAP STEP)))
(MA-N-TAP
IF (AND (TEST-GOAL DO MA PROCEDURE)
        (TEST-GOAL DO TAP STEP)
        (TEST-NOTE OPERATION SUCCESSFUL))
THEN ((DELETE-NOTE OPERATION SUCCESSFUL)
      (ADD-NOTE TYPE S-FOR SUCCESS)
      (DELETE-GOAL DO TAP STEP)
      (ADD-GOAL DO FINISH STEP)))
(MA-N-FINISHED
IF (AND (TEST-GOAL DO MA PROCEDURE)
        (TEST-GOAL DO FINISH STEP)
        (TEST-NOTE TYPE S-FOR SUCCESS))
THEN ((DELETE-NOTE TYPE S-FOR SUCCESS)
      (DELETE-GOAL DO FINISH STEP)
      (DELETE-GOAL DO MA PROCEDURE)))
```

step, and add the goal to do the step called ES-SELECT. Further detail is not necessary here; see Kieras and Polson (1985) for a more detailed description of this production system, and Anderson (1976, 1983) for discussion of production systems in general. By means of a user–device interaction simulation system (Kieras & Polson, 1985), the production rules were executed to simulate the behavior of a human interacting with a simulated control panel device; this ensured that the rules actually generated the correct sequence of actions in the various situations.

A set of production rules was written and tested in the simulation for each procedure used in the Fig. 2 experiment. Writing the production rules was done using a computer text editor, and it became obvious that once the first set of rules was written for the first procedure, then subsequent sets could be generated easily by copying the first set, doing a few substitutions, and adding a few rules when necessary. By analogy, the transfer process could consist of recognizing which new rules are identical to previously learned rules, which new rules are similar to existing rules, and which are totally new. The subject could then spend most of the training time acquiring the new rules, and merely reuse already learned rules which apply to the new situation, which appears to be possible in learning from other types of text (Johnson & Kieras, 1983).

Two basic transfer rules were defined: *identity* (from copying), and *generalization* (a form of substitution). Production rules are identical if they have the same conditions and the same actions. The original definition of the generalization transfer rule was as follows: if two production rules have the same actions, and only one point of difference in their conditions, then the rules could be generalized by replacing the differing point with a "wild card" that matches any value. For example, if the only point of difference between two rules was that one had the condition clause (TEST-GOAL DO MA PROCEDURE), and the other had (TEST-GOAL DO SA PROCEDURE), then this clause could be replaced by (TEST-GOAL DO ??? PROCEDURE), where "???" is a wild card that will match any item in that position. This generalization transfer rule was later modified as described below.

When these transfer rules were applied to the production rules for the procedure training order shown in Fig. 2, the number of new rules that needed to be added for each procedure was determined. The assumption is that the only rules that require substantial effort to learn are the completely new ones; the identical and generalizable rules should be very easy to learn, since all or almost all of their content is already known. Thus, the number of new rules in a procedure should be closely related to the difficulty of learning the procedure. The regression equation giving total training time as a function of the number of new productions is

$$\text{Time} = 162 \text{ s} + 31 \text{ s} * (\text{number of new rules})$$

which accounts for 79% of the variance among the mean training times for the 10 procedures. The predicted times are shown in Fig. 2. This supports the value of the production system analysis of transfer in the learning of procedures. However, this result was based on only one training order, and so is no more than suggestive.

EXPERIMENT

By using three different training orders, this study was designed to get a more comprehensive set of data on the relation of the production rule representation to transfer of training. The three different training orders were chosen by analyzing the production rule sets for each procedure using a simulation of the transfer process, and selecting training orders that produced substantial variation in the number of new rules in each procedure and also at each serial position. Thus, the transfer model was used to make very strong *a priori* predictions of the relative difficulty of the procedures and the shape of the training time profiles.

Overview

Each subject learned a series of 10 procedures in a fixed order. There were three different orders, chosen as described below, with a separate group of subjects for each order.

To learn each procedure, the subject first read a set of step-by-step instructions for the procedure, such as those in Tables 1 and 2, and then attempted to execute the procedure on the device. Upon making an error, the subject was immediately informed, and then began to read the instructions again. The subject was required to execute the procedure correctly three times in a row before proceeding to the next procedure. The data recorded were the reading time on each step of the instructions, the accuracy of each step while executing the procedure, and the speed and accuracy of a final retention test.

Method

Transfer simulation. A simple simulation program was written in LISP to represent the transfer process. The transfer simulation is given a series of procedures, each consisting of a set of production rules. The rules for each procedure are examined for possible transfer with the set of rules already known. The simulation reports the number of rules considered identical to existing rules, the number that could be generalized with existing rules, and the number of new rules added to the total. It updates its rule set accordingly, and then goes on to the next procedure. The final rule set was tested in the user–device interaction simulation to check that a correct rule set for all 10 procedures was generated.

The generalization criteria in the simulation were modified slightly from the original definition. Certain types of rule clauses could not be generalized. These were clauses that sequence the firing of rules (e.g., goals of the form DO STEP X), clauses that look for a particular configuration of indicator lights on the device, and clauses that operate controls on the device. Thus only clauses involving notes and goals could be generalized. The new generalization process could generalize more than one clause in the condition, and could also generalize the corresponding clauses in the action part of the production rule.

Training order conditions. The transfer simulation was used to select the three training orders for the experiment that would maximize the predicted effects. That is, either a procedure would have different predicted training times in the different orders because there were a different number of new rules to be acquired, or if the number of new rules were the same, then the procedure would be in a different serial position in the different training orders. These different orders also produced different numbers of rules defined as identical or generalized. A final constraint on the training orders were that they should be, in some sense, meaningful orders, rather than apparently random.

The selected training orders are shown in Table 5, which shows the number of new rules for each procedure in each order. In training order condition 1, all the MA command procedures are presented first, and then the SA command procedures. Within each command, normal procedures are first, non-compensatable malfunctions second, and possibly compensatable malfunctions last. Training order condition 2 is based on the idea that once the longest procedures are learned, the shorter procedures should be learned comparatively easily. Thus, the possibly compensatable malfunctions are first, noncompensatable malfunctions second, and normal procedures are last. Within these groups the MA proce-

TABLE 5
NUMBER OF NEW PRODUCTION RULES FOR EACH TRAINING ORDER CONDITION

	Training order conditions					
	1		2		3	
Serial position	Procedure	New rules added	Procedure	New rules added	Procedure	New rules added
1.	MA-NORMAL	9	MA-XMA	13	SA-NORMAL	9
2.	MA-XEB	5	MA-XMA-XSA	4	MA-NORMAL	2
3.	MA-XPB	1	SA-XSA	5	SA-XEB	5
4.	MA-XMA-XSA	4	SA-XPB	0	MA-XEB	0
5.	MA-XMA	2	MA-XEB	1	SA-XPB	4
6.	SA-NORMAL	2	MA-XPB	1	MA-XPB	1
7.	SA-XEB	0	SA-XEB	0	MA-XMA	5
8.	SA-XMA-XSA	1	SA-XMA-XSA	1	SA-XSA	0
9.	SA-XSA	3	MA-NORMAL	2	SA-XMA-XSA	1
10.	SA-XPB	0	SA-NORMAL	0	MA-XMA-XSA	0

dures are presented before SA. Training order condition 3 is based on the principle of underlying causes, even though subjects have no information on these causes. The pair of normal procedures are first, followed by each pair involved with each malfunction state. Within these pairs, SA procedures came before MA procedures.

Instruction materials. A set of step-by-step instructions were prepared for each procedure; examples appear in Tables 1 and 2. These were prepared so that each sentence in the instructions appeared to correspond to a single production rule, one for each step or action (overt or covert) involved in the procedure, and the rules were carefully compared to the instructions to ensure the correspondence was tight. An example of the correspondence can be seen by comparing Tables 1 and 4.

Apparatus. The device consisted of an actual physical control panel connected to a laboratory computer, which monitored the settings of the switches and push buttons and controlled the indicator lights accordingly. All instructions and commands to the subjects were presented on a standard video terminal positioned next to the device. A computer-assisted instruction facility was used to present all of the procedure training and the retention tests. The subject was seated in a small room at a table with the terminal and the control panel, and was observed by means of a video camera and monitor.

Subjects. Subjects were recruited through campus advertisements and were paid $5 for their participation. Subjects were randomly assigned to each of the three training order conditions. A total of 70 subjects participated in the experiment. The data of 10 subjects were discarded, leaving a total of 60 subjects, with 20 subjects in each condition. Of the 10 subjects whose data were discarded, two final subjects were discarded because their data were not needed, three subjects did not finish the training part of the experiment, one subject was discarded because of a fire alarm during the experiment, and the first four subjects were not used because of problems in the experimental software.

Design. Training order condition was a between-subjects factor, with each subject randomly assigned to one of the three training order conditions, subject to the constraint that during the experiment, approximately equal numbers were maintained in the three conditions. Each subject learned all 10 procedures in all three conditions. Subjects were also assigned by gender, so that there would be an equal number of males and females in each condition.

Instructions and procedure. The first part of the instructions familiarized the subjects with the layout and labels on the device. Subjects were then told that they would be trained in several procedures for operating the device. They were told that the goal of operating the device was to make the PFI indicator flash. Part of their training would include procedures to be performed if the device malfunctioned. They were told that for some malfunctions the PFI indicator would not flash at first, but it might be possible to change the control settings to that it would flash. This was called *compensating for a malfunction*, and it was pointed out that some malfunctions could not be compensated for. The subjects were instructed that whenever they were asked to turn the device to the *initial state,* they should set the SP switch off, the ESS selector to N, and not push any buttons.

The training for each procedure consisted of alternating *reading* and *trying* phases. In the reading phase, the subject read the procedure a single step at a time, in a self-paced reading paradigm. Then in the trying phase, the subject attempted to execute the procedure correctly. After the attempt, the subject would return to the reading phase. This process was repeated until the subject had completed three correct attempts in a row. Then the subject would commence learning the next procedure.

In the reading phase, the subject would tap the space bar to read each step on the terminal screen, which appeared as one sentence, as illustrated in Tables 1 and 2. The previous step was erased from the screen. Subjects were instructed to study each step for as long as they felt necessary. The lab computer recorded how long the subject left each step on the screen, defined as the reading time. When the subject had read all the steps in the procedure, a command, such as "Do the MA procedure," would appear on the screen and the subject would then try to perform the procedure from memory. If the subject made a mistake while attempting the procedure, the lab computer immediately sounded a buzzer, as a signal to stop trying. Then the subject was returned to the beginning of the reading phase. If the subject performed all steps correctly, the computer sounded a bell tone, and either returned to the beginning of the reading phase or went on to the next procedure if the criterion had been achieved. Throughout the procedure, the subjects were prompted by displays on the terminal screen, such as a message that they had made an error and were being returned to the reading phase.

Since some pilot subjects tended to ignore the indicators during training, the instructions included a notice that although it might seem unnecessary to pay attention to the indicator lights during training, during the testing phase at the end of the experiment, it would be necessary to rely on the pattern of indicator lights to choose the correct procedure.

After being trained to criterion in all 10 procedures subjects were instructed that they could take a short rest or break before starting the test. They were told that they would see each of the 10 procedures three times each in the test in a random order. No feedback was given during testing.

Results

Training Time

The total training time for a procedure is defined as starting when a subject begins the first reading of the first sentence of the instruction steps, until completing the last step of the last attempted execution of the procedure.

The first analysis was simply to verify the presence of gross effects of the training order on training time. An analysis of variance was performed on the total training time for each procedure in each training order condition; the means are shown in Table 6. There were main effects of training order condition and procedure, and an interaction between training order condition and procedure ($p < .05$). While female subjects were an average of 10 s faster than males on the training, this difference was not significant, and there are no significant interactions with gender.

These data can be analyzed similarly to the preliminary results shown in Fig. 2. The regression equation for mean training time as a function of the number of new rules is

$$\text{Time} = 85.3\ \text{s} + 20.2\ \text{s} * (\text{number of new rules})$$

which accounts for 69% of the variance among the 30 mean training times. Thus, at this simple level, the Fig. 2 analysis is confirmed; the number of new production rules is an excellent predictor of training time.

A more elaborate multiple regression analysis was performed in order to examine other effects in the data, and to test the

TABLE 6
MEAN TRAINING TIMES (s) FOR EACH PROCEDURE FOR THE THREE TRAINING ORDER CONDITIONS

Procedure name	Training order		
	Condition 1	Condition 2	Condition 3
MA-NORMAL	212.496	81.125	111.883
SA-NORMAL	89.863	92.814	221.958
MA-XEB	142.058	111.907	98.829
MA-XPB	109.430	96.679	108.727
MA-XMA	117.012	464.089	165.727
MA-XMA,XSA	161.478	190.291	139.679
SA-XEB	79.677	84.980	160.697
SA-XMA,XSA	86.568	99.644	95.250
SA-XSA	111.109	176.169	151.411
SA-XPB	117.109	136.013	191.817
Mean	122.727	153.371	144.598

theoretical analysis in more detail. The dependent variable was the total training time (TRTIME), giving 600 data points, one for each subject on each procedure in each condition. The major predictor variables were those provided by the transfer simulation: the number of new productions (NEW), the number of generalized production rules (GEN), and the number of identical or old production rules (OLD). Other predictor variables included the subject's mean training time for all procedures (SMEAN) to handle the within-subject design (see Pedhazur, 1982), the main effect of serial order (ORDER), and two dummy-coded variables (COND1 and COND2) to test for a main effect of condition, with condition 3 as the baseline. Since the first procedure trained appeared to require a disproportionately long time, a dummy variable, FIRST, was defined to indicate whether the procedure was the first to be trained. Two interaction variables, C1FIRST and C2FIRST, defined as the products of the dummy variables, were defined to represent the interaction of condition and first procedure.

The results of this regression analyses are shown in Figs. 3, 4, and 5, which show the predicted and observed mean times for each training order condition, and in Table 7, which shows the coefficients in the final equation that includes all variables that entered the stepwise analysis. The *F* ratios are the "*F* to remove," and so provide a test of significance of the coefficients in the final equation under the assumption that each variable was the last to enter. Thus, if a variable is nonindependent of others, the significance test is conservative. Finally, the standardized regression coefficients allow comparisons of the importance of each variable independently of the scale differences involved in the partial coefficients. About 76% of the variance in individual subject's total training time on each procedure ($N = 600$) was accounted for by the final equation.

The most important predictor variable was the number of new rules in each procedure (NEW), which alone can account for 69% of the variance, and uniquely accounts for about 47% of the variance. The partial and standardized regression coefficients for NEW are substantially larger than those for identical (OLD) rules and generalizable rules (GEN), which are very similar. Thus, each new rule adds about 19 s to the training time, in contrast to each old and generalizable rule, both of which add about 11 s. Notice that the gross number of rules is related to the overall length and difficulty

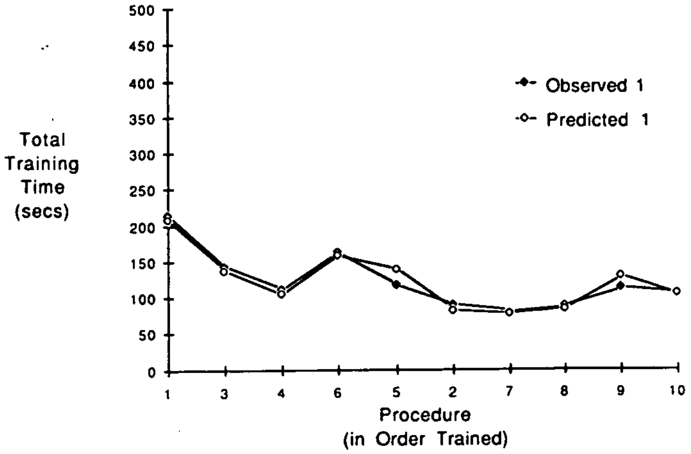

FIG. 3. Mean predicted and mean observed training times for Training Order Condition 1.

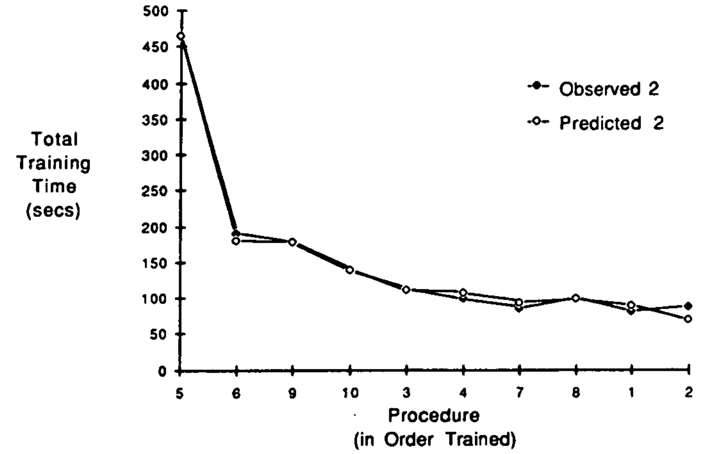

FIG. 4. Mean predicted and mean observed training times for Training Order Condition 2.

of the procedure, which is why even the number of old rules predicts the training time.

To clarify the relation of the total number of rules to the training time, an alternative regression analysis was done in which the production rule predictor variables were the *total number* of rules, the number of generalizable rules, and the number of new rules, with the same other variables. The number of steps in a procedure was perfectly correlated with the total number of rules, but essentially uncorrelated ($r < .1$) with the number of generalized rules and the number of new rules. This regression analysis yielded identical predictions and proportion of variance accounted for as the above analysis, but shows that each rule requires about 11.9 s, whether old or generalized, but each new rule requires an additional 7.8 s. Thus, during training the subject had to read and execute each step in the procedures, regardless of whether that step already had been learned, and so there is a time cost for each rule, but each *new* rule to be learned takes a substantial additional amount of time.

In addition to the substantial effects of the production rule variables, there were some learning-to-learn effects. As shown

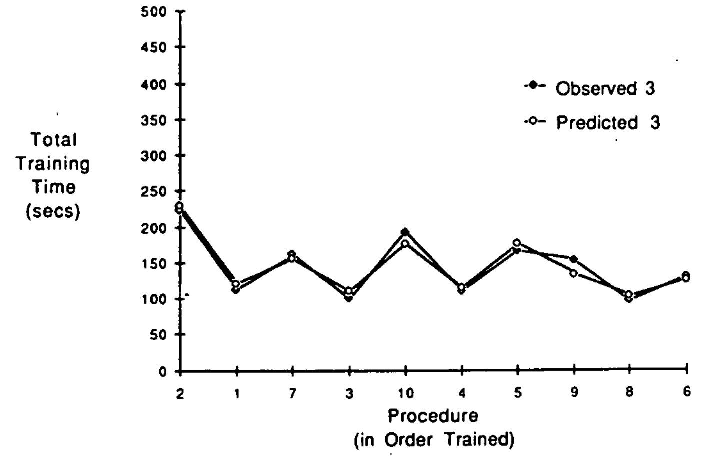

FIG. 5. Mean predicted and mean observed training times for Training Order Condition 3.

TABLE 7
Regression Analysis on Total Training Time
(N = 600, R^2 = .7623)

Variable	Final coefficient	Final standard coefficient	F
CONSTANT	−132.39		
SMEAN	1.00	.410	389.78
NEW	19.38	.662	153.54
OLD	11.82	.499	88.44
GEN	11.07	.291	51.09
C2FIRST	165.10	.324	125.04
FIRST	47.10	.155	16.04
ORDER	−3.93	−.124	18.32
COND2	−16.51	−.085	14.86

by FIRST, the first procedure involved an additional 47 s of learning time, and each procedure was learned about 4 s faster, as shown by ORDER. Of special interest is an apparent "overload" effect, shown by C2FIRST, in which the first procedure in the second training order condition took an extremely long amount of time, 165 s, to learn beyond that predicted by the number of new production rules and the other variables. This procedure was MA-XMA, shown in Table 3, which involved trying the MA setting first, then the SA setting; the first few steps have no apparent effects. The other two training orders had first procedures that were relatively simple normal operation procedures, which may have appeared obvious and natural. This sort of conceptual difficulty is clearly a matter for further research. Overall, however, this training order condition was somewhat faster than the baseline condition 3, as shown by COND2.

Despite these other effects, however, the production system variables provided by the transfer model explain the training times very well; in fact, the number of new rules alone accounts for 69% of the variance, and is a better predictor of training time on a single procedure than the subjects' individual means, which variable alone accounts for only 41% of the variance. Thus, by analyzing the procedures in terms of transfer of production rules, it is possible to account for the relative difficulty of learning the procedures with great precision.

Reading Time

The time required to read each sentence of the instructions was averaged over procedures, but classified by training trial (e.g., first reading, second reading, and so forth), and by the transfer status of the corresponding production rules (Old, Generalizable, New). Figure 6 shows these means. As is clear from the figure, there was a substantial difference in the reading times for instruction steps depending on the transfer status of the corresponding production rule. The reading times for generalizable and old rules were almost identical, but reading times for new rules were much longer for the first few readings. A key result is that this difference appears *on the first reading,* meaning that subjects can immediately distinguish whether a sentence corresponds to a new rule or to a known one, and can immediately govern their reading and study times accordingly. The difference between reading times on the first trial between New and Generalized is strongly significant (N (New) = 1567, N (Generalized) = 900, z = 3.51, $p < .01$).

A second question about the reading times is how they relate to the acquisition of individual production rules. Figure 7 shows the mean reading times for individual sentences plotted in terms of *relative trial to mastery.* The trial of mastery of a sentence was defined as the reading trial after which the subject executed the corresponding step in the procedure correctly for all trials thereafter. Thus the trial shown as −1 is the trial prior to the one upon which subjects apparently mastered the step. The figure shows the mean reading times for sentences classified by whether the corresponding production rule was new, generalized, or identical. A method was developed to eliminate the extreme

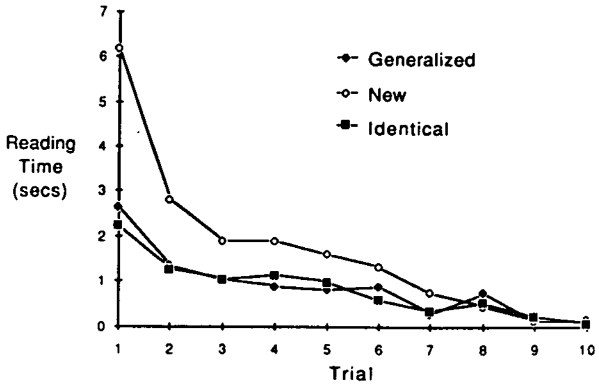

FIG. 6. Mean reading times for instruction sentences as a function of reading trial and the transfer status of the corresponding rule.

outliers in these data, whose distribution was radically skewed. Reading times were dropped that were larger than a criterion value set by Tchebycheff's inequality (Hays, 1981) to ensure that less than 10% of the data would be dropped regardless of the underlying distribution; only 2% of the data points were eliminated. As shown in Figure 7, new rule sentences are read for a long time, until subjects are able to execute the corresponding step correctly, whereupon their reading time drops quickly. Thus, acquisition of the rule, defined in terms of error-free execution, corresponds to a sudden drop in reading time.

In order to determine the significance of the apparent effects, these data were subjected to a fairly complex regression analysis, summarized in Table 8. The reading time depends on the subject's mean (SMEAN) and the number of WORDS in the sentence, and there is a simple main effect of relative trial number (RELTRL), corresponding to the overall downward trend. There is an apparent practice effect, because sentences whose steps are mastered later, as shown by larger values of MASTRL, are read for less time. The key results are NEW sentences are read longer (overall, by .723 s) than Identical or Generalizable, which are almost the same, corresponding to the Figure 6 data. Sentences before mastery are read about .739 s longer than after, as shown by BEFMAS. The effect is mostly due to the New sentences; the interaction variable, BMNEW, shows

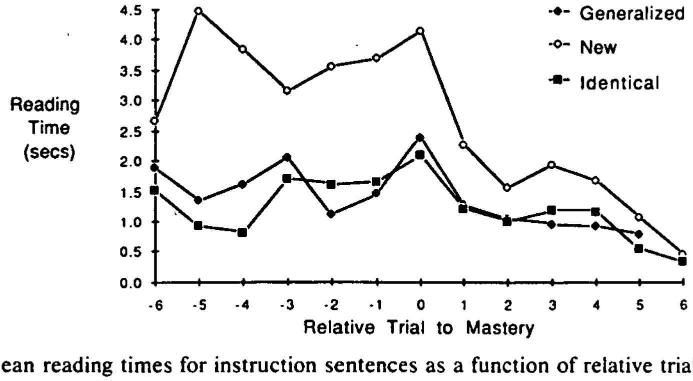

FIG. 7. Mean reading times for instruction sentences as a function of relative trial of mastery and transfer status.

TABLE 8
REGRESSION ANALYSIS ON INDIVIDUAL SENTENCE READING TIMES ($N = 21{,}449$, $R^2 = .40$)

Variable	Final coefficient	Final standard coefficient	*F*
CONSTANT	−.557		
WORDS	.069	.188	1223.60
SMEAN	.844	.316	3508.06
MASTRL	−.267	−.153	655.46
NEW	.723	.188	798.56
RELTRL	−.182	−.132	224.06
BEFMAS	.739	.210	599.37
BMNEW	1.247	.208	831.06

that NEW sentences before mastery require an additional 1.297 s. Thus, sentences that state new rules are studied until the corresponding rules are mastered, whereupon they are studied for much less time. Sentences that state old rules are never read for very long. This implies that subjects can monitor whether they have acquired the production rule stated by a sentence, and regulate their reading and studying accordingly.

Retention Effects

A full description of the retention data will not be presented here, for reasons that will be explained. The average proportion correct on the retention test was .66; this fairly low value suggests that the training paradigm did not train the procedures very well. There were substantial differences in retention between different procedures, ranging from .89 correct to .36, and a powerful interaction effect between procedure and training order condition. For example, one procedure was recalled with .53 accuracy in one training order, and with only .18 accuracy in a different training order, while other procedures varied little. Since the transfer model predicts that the learner will have the same rules after learning all procedures, regardless of the training order, these differences in retention accuracy are intriguing.

A detailed examination of the errors made during the retention test suggests an explanation of the retention results in terms of the classical interference theory of forgetting. Almost all of the execution errors can be attributed to a failure of the correct rule to fire at that point in the procedure, while a different, incorrect rule fires instead (similar to response competition). The interfering rule almost always has a condition that is very similar to the correct one, differing, for example, only in whether a certain indicator light is on rather than off (a stimulus similarity effect). There is a suggestion in the retention data that the probability of an incorrect rule interfering with a correct one depends on how many times each of the rules was used in different procedures (a degree of learning effect), and which rule was learned first (proactive versus retroactive interference).

Thus, a good first approximation to the retention results is that individual production rules in a series of procedures behave much like paired-associate items in list learning. This would be expected, given that the conditions of learning and retention are similar to standard verbal learning paradigms. The large size of the interference effects indicates that the amount of training was fairly small; clearly, the subjects did not have the benefit of overlearning.

Going into more detail on these retention results would not be useful because most of the errors can be attributed to only a small number of rules, and these rules are covert in the sense of specifying an internal decision, such as Step 5 in Table 2, rather than producing an overt action. Since these covert interfering rules are more similar to each other than the overt rules that could produce the errors, it seems reasonable to blame the covert rules for the interference effects. But this inference from overt actions back to covert rules is undesirable; it would be better to explore the retention effects with procedures and training orders in which the interfering rules can be confidently identified.

CONCLUSIONS

Production Rule Analysis

A basic conclusion is that production rules, as a way to represent procedural knowledge, can provide a detailed account of important learning processes. This supports the approach presented by Kieras and Polson (1985), who suggest that the production rule theory of skill acquisition is useful for practical applications such as the design of computer user interfaces. That there are other phenomena involved, such as the "overload" described above, is revealed by the analysis as well, although further work is required to characterize these effects. Thus, by providing precise characterization and quantification of effects such as transfer of training, the analysis of procedural knowledge in terms of production rules may lead to many scientific and practical benefits.

Comprehension Processes in Procedure Acquisition

The reading time data reveal some important features of how procedures are acquired from text. The basic conclusion is that the transferability of a piece of procedural knowledge is determined by a comprehension process that is required until the rule can be executed successfully, and not thereafter. Hence, the initial acquisition of the procedure is a comprehension process, not a skill-learning process.

These results are significant for the theory of skill acquisition as formulated by Anderson (1982, 1983). The transfer process defined here is similar to some of Anderson's compilation and tuning processes. In particular, the generalization case of transfer in the present model resembles Anderson's generalization mechanism, which is defined in terms of operations on procedural representations. These representations are constructed as a by-product of the activity of general interpretive procedures that are driven by an initial declarative encoding. However, in these results, rules are being compared, modified, and constructed very rapidly, and apparently before they exist in a procedural form. For example, as Fig. 6 shows, a generalization process can apparently occur on the first reading, and is almost as fast as recognizing an identical rule. Thus generalization can appear not only in compilation and tuning, but also in the form of operations on the declarative representation of a production rule. Relating new and old rules could be done by processes similar to those proposed for macroprocesses in comprehension (e.g., Kieras, 1982), which can compare, modify, and construct complex propositional representations while reading is going on.

Thus, in the process of acquiring procedures from text, complex comprehension processes that construct the initial declarative form of the production rules can play a major role early in learning by taking advantage of prior knowledge; the compilation and tuning processes govern learning once the correct declarative representation of the rules is in place.

An Outline of a Process Model

As a way of making the above discussion more precise, an outline of a process model for acquiring a procedure from written instructions, illustrated in Fig. 8, will be described. Assuming that the input text is processed one sentence at a time, the basic comprehension processes perform parsing and simple referential and semantic analysis to convert the input sentence into a propositional representation in working memory. Various *procedure comprehension* processes then act on the sentence content to construct a declarative representation fnr that step in the procedure. This representation is essentially isomorphic in content to the corresponding production rule, but consists of a propositional representation, along the familiar lines of Kintsch (1974) or Anderson and Bower (1973), rather than being a true production rule in procedural knowledge form. As with

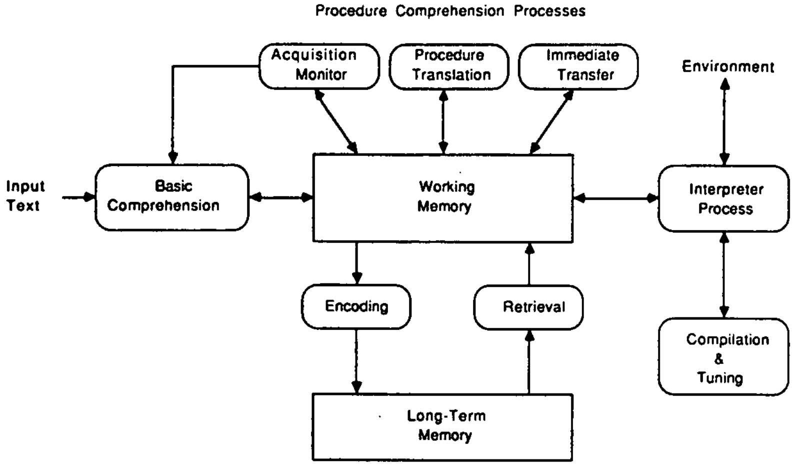

FIG. 8. An outline of a process model for acquiring a procedure from text instructions.

any other declarative knowledge, this representation of the procedure can be encoded into, and retrieved from, long-term memory.

The interpreter process accesses the declarative representation and executes the procedure by interacting with the environment (e.g., the piece of equipment). This execution will be successful as soon as the declarative representation is correct and complete and can be successfully retrieved from either working memory or long-term memory. With repeated execution, Anderson's (1982, 1983) compilation and tuning processes would respond to the activity of the interpreter process by constructing a executable procedural representation of the procedure, resulting in faster and more reliable execution as practice continues.

In order to execute the entire procedure correctly in this experiment, the reader must successfully construct and have available the declarative representation for the entire procedure. For a short procedure, it seems reasonable that the entire procedure representation could be constructed and maintained in working memory. The overload effect discussed above could be due to this first procedure being too long to fit into working memory all at once, meaning that the reader will have to get some of it encoded into long-term memory before the procedure can be executed successfully. Another possibility is that if the procedure is apparently meaningless, the reader may not be able to chunk the procedure steps, or to use mnemonic strategies to allow the procedure to fit in working memory. More detailed experiments and the construction of a simulation model would clarify this effect.

The interference effects in retention mentioned above could be due to classical memory interference effects involved with either the declarative representation in long-term memory, or the procedural representation. The few training trials involved in this experiment are consistent with at least part of the procedure still being in declarative form.

The procedure comprehension processes can be described in more detail. The first, *procedure translation,* translates the semantic content of a step-by-step instruction sentence into the declarative representation of a production rule. As with other forms of comprehension, this involves heavy use of implicit information. Notice that rarely is an individual step sentence stated in the IF–THEN form of a production rule, even though the results show that this appears to be an accurate characterization of its implicit content. Normally, the only time a step is phrased in a conditional form is when the step itself involves a decision, such as Steps 4 and 5 in Table 2. Usually the instruction sentences consist of just the name (number) of the step, and an action to be performed. What is implicit is the information on what context the action should be performed in, such as the overall goal or name of the procedure, and the specifics of what step is done next. This information is implied by the arrangement of the sentences and the semantics of the step names. The translation process takes the step name and the stated action and combines it with the overall procedure name to produce a fully specified condition–action pair that has the conditions and actions to properly sequence the rule in the procedure as a whole. The reader can get some sense of what would be involved by comparing the actual explicit content of the sentences in Table 1 with the content of the rules shown in Table 4.

A second procedure comprehension process is the *immediate transfer* mechanism described above. Once the translation process has constructed the representation of a rule in working memory, it can be compared to the rules already acquired. If it is a new rule, it must be maintained in working memory and encoded into long-term memory, requiring the extra time shown by the training time and reading time effects. If the rule is identical to an old rule, or similar to an old rule that can be modified, it does not need to be represented in working memory, nor does it need to be encoded into long-term memory. Notice that the modifications to existing rule representations are very small, perhaps consisting of changing only one or two propositions, meaning that the time to generalize an old rule would be similar to the time to recognize an identical rule, consistent with the results.

A third procedure comprehension process is an *acquisition monitoring* process, which is implied by the reading time for a sentence dropping substantially once the rule for the step is acquired. There must be a process which monitors the execution success of each rule in the declarative representation, and determines which sentences must be studied again, and which can be skipped. Since repeated trials might be necessary before a new rule is fully encoded, the reading time might stay high for several trials, as shown by the plateau before mastery in Fig. 7. The gradual decline in reading times for new sentences shown in Fig. 6 is the familiar consequence of the distribution of downward steps over trials. The downward trend in time after a rule is mastered, shown in Fig. 7, can be attributed to the declarative representation becoming easier to retrieve with practice, or to the development of skill at recognizing the sentence as already known.

The components and processes outlined in Fig. 8 are based on familiar mechanisms, and some of them have already been implemented in cognitive simulation models. It would be a straightforward task to construct a complete simulation model for acquiring procedures from text, and the adequacy of these ideas could be tested by comparing the model against time data along the lines suggested in Kieras (1984). Defining the procedure comprehension processes with a simulation model would help characterize the linguistic conventions for procedural text, and the conditions under which procedures can be easily acquired from text. Such knowledge could then be applied to the practical problem of how instructions could be made more comprehensible and useful.

REFERENCES

ANDERSON, J. R. (1976). *Language, memory, and thought*. Hillsdale, NJ: Erlbaum.

ANDERSON, J. R. (1982). Acquisition of cognitive skill. *Psychological Review*, 89, 369–406.

ANDERSON, J. R. (1983). *The architecture of cognition*. Cambridge, MA: Harvard Univ. Press.

ANDERSON, J. R., & BOWER, G. H. (1973). *Human associative memory*. Washington, DC: Winston.

HAYS, W. L. (1981). *Statistics* (3rd ed.). New York: Holt, Rinehart & Winston.

JOHNSON, W., & KIERAS, D. (1983). Representation-saving effects of prior knowledge in memory for simple technical prose. *Memory & Cognition*, **11**, 456–466.

KIERAS, D. E. (1982). A model of reader strategy for abstracting main ideas from simple technical prose. *Text*, **2**, 47–82.

KIERAS, D. E. (1984). A method for comparing a simulation model to reading time data. In D. Kieras & M. Just (Eds.), *New methods in reading comprehension research*. Hillsdale, NJ: Erlbaum.

KIERAS, D. E., & BOVAIR, S. (1983). *The role of a mental model in learning to operate a device* (Technical Report No. 13, UARZ/DP/TR-83/ONR-13). University of Arizona, Department of Psychology.

KIERAS, D. E., & BOVAIR, S. (1984). The role of a mental model in learning to operate a device. *Cognitive Science*, **8**, 255–273.

KIERAS, D., & BOVAIR, S. (1985). *The acquisition of procedures from text: A production-system analysis of transfer of training* (Technical Report No. 16, TR-85/ONR-16). University of Michigan.

KIERAS, D. E., & POLSON, P. G. (1985). An approach to the formal analysis of user complexity. *International Journal of Man–Machine Studies*, **22**, 365–394.

KINTSCH, W. (1974). *The representation of meaning in memory*. Hillsdale, NJ: Erlbaum.

PEDHAZUR, E. J. (1982). *Multiple regression in behavioral research* (2nd ed.). New York: Holt, Rinehart & Winston.

(Received February 26, 1986)
(Revision received May 19, 1986)