DOCUMENT RESUME

ED 111 396 IR 002 452

AUTHOR Mayer, Richard E.
TITLE Instructional Variables in Meaningful Learning of Computer Programming.
PUB DATE Apr 75
NOTE 4p.; Paper presented at Annual Meeting of the American Educational Research Association (Washington, D. C., March 30-April 3, 1975)

EDRS PRICE MF-$0.76 HC-$1.58 Plus Postage
DESCRIPTORS Computer Programs; *Computer Science Education; Educational Research; *Learning Processes; Models; *Programing Languages; Research Projects; Teaching Techniques
IDENTIFIERS Ausubel (David); Mathemagenic Activities; Rothkopf (E Z)

ABSTRACT

Some 120 undergraduate students participated in experiments to learn how novice computer programers learn to interact with the computer. Two instructional booklets were used: A "rule" bookl . consisted of definitions and examples of seven modified FORTRAN statements and appropriate grammar rules; the "model" booklet was identical with the addition of a page describing computers in familiar terms. Two types of questions were used: generative and interpretive. Three experiments were conducted using various combinations of booklets and questions. Findings support the idea that a meaningful set of familiar experiences prior to instruction may result in a learning process of assimilating new material and organizing it in the context of that set. (JY)

Richard E. Mayer
Department of Psychology
Indiana University
Bloomington, Indiana 47401

Title: Instructional Variables in Meaningful Learning of Computer Programming

Author: Richard E. Mayer, Indiana University

Objectives: Although it seems clear that computer technology will play an increasing role in education, little is presently known concerning how novice programmers learn to interact with a computer (Weinberg, 1971; Miller, 1972), nor how to develop technical instruction which results in meaningful learning (Mayer, 1972). The present research attempted to provide modest information in these two areas.

Theoretical: Two theoretical questions concerning human learning and cognitive processes are addressed. (1) The first concerns Ausubel's (1968) concept of "meaningful learning set" and the idea that meaningful learning consists of assimilating new material to familiar concepts already existing in memory. According to this view, an "advance organizer" which presents a concrete model of the computer in familiar terms (Model Group) may provide a meaningful learning set which can be used to interpret and encode subsequent facts concerning a particular computer programming language; similar learners presented with the same facts but without pre-exposure to the model (Rule Group) may lack a meaningful learning set and thus encode the material in a more piecemeal, rote manner. In contrast to this view of learning as integrating new and old knowledge, is a more straightforward idea that learning involves adding pieces of information, facts and skills to memory. The former idea predicts that Model learners and Rule learners will display different patterns of post-learning competencies since they encoded the material in different ways, while the latter view predicts that since both groups received the same basic facts and principles no such difference should be found.

(2) A second question, derived from Rothkopf's (1970) concept of "mathemagenic activities", concerns the idea that the type of questions subjects are asked to solve as practice during instruction serves as a sort of attention director which reinforces certain aspects of instruction and around which presented material is organized. Questions requiring serious thought may encourage deeper, more meaningful encoding, while question emphasizing direct application of presented material may encourage concentration on memorizing the basic facts. According to this view, practice exercises on understanding and interpreting written programs (Interpretation Questions) may reinforce a broader encoding of the material since subjects must interact more strongly with the new concepts, while practice exercises on straightforward application of rules for writing programs (Generation Questions) involve less activity and hence more attention to rote facts. This view predicts that Model learners will benefit most from Generation practice and Rule learners from Interpretation practice since these emphasize new material not emphasized in instruction. A contrasting view that practice serves only to reinforce learning established in instruction contradicts this prediction.

Method: Two instructional booklets were used. The Rule booklet consisted of seven pages, each devoted to a definition and example of each of seven modified FORTRAN statements (READ, WRITE, GO TO, IF, STOP, Counter Set, Arithmetic) and appropriate grammar rules (e.g., memory addresses, pointer labels, formating). The Model booklet presented the same seven pages but began with a page describing the computer in familiar terms -- e.g., the memory as an erasable scoreboard, the control program as a shopping list with a pointer, the input tray as a ticket window, etc. -- and each of the following seven pages included hints on how the statement related to the model.

Two types of questions were used. Generation questions stated a problem, e.g., "Given a card is in the input tray, write a program to print out double the number on the card." and asked the subject to write a program to solve it. Interpretation questions stated a program such as,

```
READ (A1)
A1=A1*2
WRITE (A1)
STOP
```

and asked the subject to specify the input and output conditions and tell what problem the program would solve. As well as linear programs (shown above), questions dealt with single statements ("Given a card is in the input tray write a program to have the number on that card store in memory space A6."), and questions about looping programs ("Given a pile of cards is in the input tray, write a program to count how many cards there are before you get to one with a 99 on it.").

In all three experiments, subjects read a booklet which either included the model or did not. In Experiment 1, subjects answered, with feedback, practice items involving both generation and interpretation of non-looping programs, and then took a test involving generation and interpretation of looping programs. In Experiment 2, subjects answered, with feedback, practice items involving only interpretation of non-looping programs or only generation of non-looping programs, and then took a test involving generation and interpretation of statements, non-looping and looping programs. In Experiment 3, subjects practiced without feedback on both generation and interpretation of non-looping programs or received no practice at all, and then all subjects took a test as in Experiment 2.

Data Source: The subjects were 120 Indiana University students who participated in the experiment in order to fulfill a requirement of their introductory psychology course.

Results: In Experiment I, an analysis of variance revealed a Model x Question Type interaction ($p < .025$) in which Rule subjects excelled on question items requiring generation of programs (46% correct) relative to Model subjects (27%) but Model subjects outperformed Rule subjects on interpretation items (27% to 11% correct respectively). A similar pattern of Model x Question Type interaction ($p < .05$) was obtained in Experiment 3, with Rule subjects outperforming Model subjects on posttest items involving non-looping generation of programs (60% to 50% correct, respectively) and Model subjects outperformed Rule subjects on interpretation of non-looping programs (62% to 38% correct, respectively) and on generating looping programs (30% to 12% correct, respectively). In Experiment 2, there was Model x Type of Practice interaction ($p < .025$) with generation practice increasing overall test scores of Model subjects most (62% correct with generation and 40% correct with interpretation practice) and interpretation practice helping Rule subjects most (50% correct with generation and 56% correct with interpretation practice). Practice had no effect in Experiment 3.

Implications: These findings support the idea that providing learners with a meaningful set of familiar experiences prior to instruction may result in a learning process of assimilating new material to and organizing it in the context of that set -- what we have called building "external connections" (Mayer & Greeno, 1972). Subjects exposed to the same material, but lacking a rich set of existing experiences, must encode the material without relating it to other ideas (low "external connections") but probably build structures which retain more of the detail of the material exactly as it was presented -- what we have called building strong "internal connections" (Mayer & Greeno, 1972). This distinction is consistent with the learning outcomes observed for the two groups: subjects who had access to a meaningful learning set developed learning outcomes which supported interpretation and transfer to different situations; subjects who lacked a rich assimilative set developed learning outcomes which best supported direct application of learned rules for writing simple programs and statements. When subjects have already had some experience in a certain area the importance of providing a "model" prior to instruction is certainly diminished and may actually conflict with a learner's established knowledge; however, in a situation such as the present experiment where no subjects had had any prior experience with computers at all, the need to explicitly present a set of familiar experiences prior to instruction is significant. Practice items seem to have had the effect of directing the learners attention to areas which were not emphasized in instruction. In other words, if subjects were given instructions which emphasized the building of external connections (Model) then practice in building internal connections (Generation) seems most helpful; while subjects given instruction emphasizing internal connections (Rule) excelled most if given practice in building external connections (Interpretation).

References

Ausubel, D. P. Educational psychology; a cognitive approach. New York: Holt, Reinhart & Winston, 1968.

Mayer, R. E. Dimensions of learning to solve problems. University of Michigan: Human Performance Center, Memo Report No. 15, 1972.

Mayer, R. E. and Greeno, J. G. Structural differences between learning outcomes produced by different instructional methods. Journal of Educational Psychology, 1972, 63, 165-172.

Miller, L. Computer programming by non-programmers. IBM Technical Report, 1973.

Rothkopf, E. Z. The concept of mathemagenic activities. Review of Educational Research, 1970, 40, 325-336.

Weinberg, G. M. The psychology of computer programming. New York: Van Nostrand, 1971.

Footnote

This research was supported by Grant EC-44020 from the National Science Foundation to the author. I appreciate the support provided by Frank Restle and the Cognitive Institute of Indiana University.