The Development of Self-Regulation Skills Through the Modeling and Structuring of Computer Programming

☐ Anne-Marie Armstrong

Anne-Marie Armstrong is at the Educational Research and Development Center, University of West Florida, Pensacola, FL 32514.

The development and enhancement of self-regulation, learning to learn, and adaptive problem solving are predicated on the ability to access and engage one's metacognitive skills. In this study, 3rd, 5th, and 8th grade students were given a series of increasingly more complex tasks which they had to perform themselves and then "teach" the computer to perform. The computer became the receiver of the students' inputs and strategies and served as a model and structure on which to gain access to their own personal problem solving strategies. Students analyzed, criticized, adapted, and changed those strategies as needed. Proficiency increased and strategies were more easily adopted and adapted to other tasks and problems. Completion of tasks increased significantly and nonproductive steps (errors) and number of trials (redos) decreased. There were no differences by sex except in the types of errors made. Evidence of self-regulation development was also shown in the types of questioning used by students.

☐ Generally, lessons in today's classrooms do not include or utilize the metacognitive concept of self-regulation or learning to learn. Seldom are students asked to examine what they know and how they know it. Learning is usually teacher directed and the regulation is "outside." Solutions and answers can be "welded to the form and context in which . . . acquired" (Brown, 1987, p. 72). Given the rapidly changing environment of the world, this type of learning no longer prepares the young to assume their responsibilities. It does not provide a basis for adaptive problem solving which involves careful examining of personal and organizational behavior and the development of workable alternatives.

Self-regulation, i.e., learning to learn, has two necessary components: metaknowledge and metaprocessing. Metaknowledge is knowing what one knows, or concept knowledge. Metaprocessing is knowing how one knows. It can be automatic processing or controlled processing. Automatic processing is possible without accessing or reconstructing the basic steps involved in the performance of the "how." It requires little attention to individual steps or processes. It is analogous to an experienced driver navigating a familiar route. Controlled processing is the conscious accessing and control of the personal strategies used to perform a task or solve a problem. Controlled processing may contain subparts which are automatic. For example, the steps used in addition are usually automatic when planning a dinner party. But addition can also be controlled processing when it is broken down into its basic counting components.

Switching from automatic processing to controlled processing is a useful skill both for solving new problems and for finding alternative solutions to existing problems. The driver who automatically turns left at a familiar crossroads must switch to controlled processing when a roadblock forces him or her to go a different route. Otherwise the driver would sit at the crossroads until the roadblock was removed.

Children meet roadblocks in learning situations when their automatic processing is not sufficient to accomplish a task. However, not all children know that they can switch to controlled processing. Instead, many depend on outside or external regulation.

The study described in this article was undertaken as a pilot for the development of a mini-course which teachers could easily use to enhance students' self-regulation skills and foster adaptive problem solving. The study was structured so that it required the articulation of both concept knowledge and process knowledge, and outside regulation was kept to a minimum. The computer was used as the receiver of the students' problem-solving instructions in order to create situations where children would be unable to rely on their automatic processing. Because the computer programmer used the BASIC language to program the students' strategies, it became the structure around which the controlled processing would take place. After the instructions were converted into computer programs, they were run and reality testing was used to reveal errors and misperceptions.

The question investigated was whether or not providing a model and structure of self-regulation skills via "teaching" a computer to solve problems or perform tasks would increase the amount of tasks completed and/or problems solved and would decrease the error and trial (attempt) rate.

Ideally, teaching promotes self-regulation, and successful teachers gradually cede their direction and power to the learners. In reality, children are seldom asked to articulate the process or plan when they are performing tasks in the classroom. Hence, they cannot list the discrete steps of their strategy when asked and do not fully access the metacognitive or control functions of the thinking process. In other words, they do not have access to the "how" of learning. A method which has been used to develop access to these control processes is having one student write a plan which another student follows step by step until each step is identified and the purpose or result of the plan is attained. The flaw in this method is that the second student might fill in gaps with his or her own strategies or correct logic unconsciously. The reasoning and the plan, then, might still remain unclear to the first student.

Dewey's (1933) concepts of reflective thought as opposed to a random or automatic coursing of thoughts through the mind is very similar to controlled processing.

> Reflection involves not simply a sequence of ideas, but a *con*-sequence—a consecutive ordering in such a way that each determines the next as its proper outcome, while each outcome in turn leans back on, or refers to, its predecessors. The successive portions of a reflective thought grow out of one another and support one another; they do not come and go in a medley. Each phase is a step from something to something—technically speaking, it is a *term* of thought. Each term leaves a deposit that is utilized in the next term. The stream or flow becomes a train or chain. There are in any reflective thought definite units that are linked together so that there is a sustained movement to a common end (pp. 4–5).

Kelly (1955) recognized the fact that most people have a personal model for problem solving but are unaware of their use of those models. Research in artificial intelligence has brought us in closer touch with our individual models and the models of others. Research on differences in problem solving between experts and novices usually begins with the articulation of the thought processes or plans of the individual during the course of the trial or task.

Computer programming is a simulation of

human controlled processing. Programming a computer involves developing a series of steps or directions which the computer must follow in order to perform a certain task. Using top-down programming, one can specify an outcome, design an approach for reaching the outcome, and then evaluate the design's efficacy in light of the desired outcome versus the "real" outcome. It is Dewey's (1933) "chain of reflective thought." The comparable processing or control skills of problem solving are planning, error monitoring, reality testing, and re-evaluation (Brown & DeLoache, 1978). Modeling the machine which models the human provides a structure within which control processes can be understood and provides a vocabulary which can be used to articulate them.

Flavell (1981) contends that cognitive monitoring (control) is a necessary step before there can be any integration and transfer of knowledge, and that cognitive monitoring is the key to critical thinking. His model of cognitive monitoring closely resembles the computer model because it requires describing the problem or issue, estimating the outcome or end product, deciding upon a strategy and implementing it, monitoring the outcome, and evaluating the process. Wellman (1982), in addressing the factors contributing to the development of the awareness of cognitive processes, states that the activities in which a child engages increase his or her *control* over the distinct processes. The extent of information processing burdens placed on the learner affect high- and low-ability students in different ways. Computer programming relieves the burden of information processing for the low-ability student through modeling, and provides challenge and complexity to a high-ability student through versatility and multiple solutions (Madinach & Fisher, 1983).

Using the computer as a model for information processing has been advocated by many researchers. The computer has been and is increasingly modeled on human intelligence systems. Papert (1980) sees learners deliberately learning to imitate mechanical thinking in order to "articulate what mechanical thinking is and what it is not" (p. 27). Hartly (1980) recommends that students try solutions in a computer program format because it requires that the student give a "precise statement in a step-by-step solution" (p. 147). Early (1975) also recommends the use of the computer model of input, processing, output, and feedback to understand "the dynamic relationship among modalities" (p. 22) or channels between the child and his environment. Recently, Pea (1985) advocated more research into the effects of cognitive trace systems, i.e., systems that externalize the products of the mind and leave a trace "of where one has been in an episode of problem solving" (p. 85). The computer's capacities represent practical means of testing whether cognitive trace systems will make developmental contributions to human reasoning.

METHOD

Subjects

Fifty-two students, 27 boys and 25 girls in grades 3, 5, and 8, in a local school district participated in this study, which consisted of nine 50-minute sessions over a two-week period, on site, with approximately ten students in each session. The students were randomly selected by the schools' curriculum coordinators. Students who were already being pulled out for other purposes were usually not included. Parental permission was obtained.

Procedure

In the initial session, the entire group was introduced to ten things a computer can do, i.e., print, perform mathematical functions, repeat tasks, choose among alternatives, store, retrieve and display data, check for equality, change values, follow instructions in a given order, and pause while awaiting further instructions. In the second through ninth sessions, the students were given a choice of completing specific tasks individually or with partners. These arrangements were very fluid. In work sessions two through nine, the students constantly changed from individual work to dyads and triads and then back to individual work. Any task completed by a group was counted only once for completion and error rate.

The instructions for each task were the same and were printed on separate sheets in the students' workbooks along with the tasks. Following is a facsimile of Task I and its instructions.

TASK I

1. On this sheet, write out your plan for completing the task.
2. Complete the task on this work sheet.
3. Write out a plan or instructions for the computer to perform the same task.
4. Give your instructions to the coder.
5. If the computer did not do what you wanted it to do, find the problem and correct it.
6. Attach the printout of the result and a list of the instructions you gave the computer.

Print the following exactly as shown:
GREENGIRAFFES
YELLOW YAKS

Below the task, sufficient space was provided for both the students' performances of the tasks and the two sets of students' written instructions or plans. The coder was a university student who had been trained to convert those instructions or plans into BASIC. The coder always provided a hard copy of the results along with a program list to the students. Copies of these along with the written instructions were kept for analysis. The tasks were designed to progress from simple, single-procedure tasks to complex tasks requiring repetitive procedures or tasks which have two or more distinct procedures. Half of the tasks were school-oriented (e.g., "print the following problems and their answers") and half were not (e.g., "using stars instead of lines, draw a shape as close as possible to a square"). The students were not informed of the purpose of the study. They were told that it was a course in which they would learn more about computers. Student performance was tracked and scored with protocol sheets, tape recordings, student handwork, and the program lists and results printed directly from the computer.

The protocol sheet was designed to keep track of the problem-solving steps articulated (written) by the individual student. When students orally changed their instructions, they had to note that change on the program plan. The protocol sheets were designed to include the problem-solving steps which are comparable to the metacognitive skills of planning, error monitoring, reality testing, and re-evaluation. There were five categories: recognition, planning, organizing, monitoring, and implementing. Each category was further divided into either a "productive step" or a "nonproductive step" according to its contribution to the performance of the task. The protocol sheets were coded after the students' plans were tested by running them as a program on the computer. Therefore, the successful completion or outcome, which was judged by the student as matching or not matching the task, was known. For example, in Task I, if the student's plan stated that the computer should print the words YELLOW YAKS directly below or on a second line below GREENGIRAFFES, this was a productive step under recognition, planning, and organizing. If, however, the student failed to state the exact location of the letters or the form they were to take, and the outcome differed from the printed workbook version, a nonproductive step, unspecific or unclear, would also be noted. On the second trial or attempt it could then be noted how the student monitored his plan and where changes were made. Protocol sheets were filled out for every attempt. Students moved from one task to the next at the point when they *said* that they had completed the task. The task was coded as completed when the outcome matched the task. The process could differ as long as the outcome fulfilled the requirements set forth for that task. Alternative solutions were frequently used. Individual performance varied.

RESULTS AND DISCUSSION

Nonproductive steps (or error type), tasks attempted, tasks completed, task sequence, and the debugging and monitoring strategies employed were recorded. These data were analyzed by sex, grade level, and task type.

It was hypothesized that the students would improve their completion rate, employ more productive steps, and have fewer nonproductive steps as they progressed through the tasks.

Completion rates for the second group of tasks, which were more complex than the first group in the sequence, were significantly greater. Seventy-eight percent of the subjects completed these more complex tasks, while only 58% completed the first group of simpler tasks ($Z = 2.5$, $p < .05$). Also, the average number of attempts at tasks (trials) as well as the average number of nonproductive steps were lower for the second group of tasks (see Table 1).

In addition to supporting the concept of providing models and structures for problem solving and task completion, the results also indicate a growth in the development of self-regulation skills for the students. This growth of self-regulation skills is exemplified in the tape recordings of the sessions. The questions asked by the students at the beginning of the two-week session differed from those asked at the end of the session. The most frequently asked question during sessions one and two was, "Is this right?" This question was directed at the experimenter with no attempt at self-determination or reality testing. It was a non-question. In contrast, the following question was recorded near the end of the second week: "Are you trying to say if you put it in quotations then it will give you the problem and the answer, but if you do it without quotations then it is going to only give you the answer?"

Another example occurred while students were performing the second task, which was to print a statement 25 times. The students were told that they had to really explain to the computer what 25 meant. Students began using some very simplified definitions (a two and a five). However, it was not sufficient for the computer. A breakthrough came when a student graphically represented 25 as 25 separate lines on his paper. The counting process had become so automatic to these third-, fifth-, and eighth-graders that it took them a long time to realize that mathematics and numbers were simply a representation of the count.

Detailed analysis of the types of errors and their occurrences was also revealing. Errors did not linearly decrease as students progressed through the tasks. Instead, there seemed to be individual patterns. Students made their breakthroughs of thought at differing points and, at times, the evidence of restructuring did not occur until certain other tasks were attempted. One student continually used an uncontrolled GOTO statement in almost every instruction to the computer even though it was usually inefficient and at times detrimental. In a flash of understanding, she eventually exclaimed, "But I don't even need that [the GOTO] in here!" Additional analysis of error types or nonproductive steps, tasks attempted, and task completion follows.

Table 2 shows the types and frequency of nonproductive steps by sex. By far the most frequent nonproductive step was misidentification. A misidentification was defined as a nonproductive step which affected the recognition step of the problem-solving process.

TABLE 1 □ Completion Rate, Average Number of Attempts, and Average Number of Nonproductive Steps

	Tasks I–IV	*Tasks V, VI, VII, VIII & X*
Completion Rate	58%	78%*
Average Number of Attempts	1.36	1.26
Average Number of Nonproductive Steps	2.34	1.51

*$p < .05$

TABLE 2 □ Types and Frequency of Nonproductive Steps

	BOYS		GIRLS	
	N	*Avg.*	*N*	*Avg.*
Identification	191	7.07	157	6.28
Planning	134	4.96	114	4.56
Organizing	77	2.85	79	3.16
Changed Task	8	.30	13	.52

That is, the directions given were either not related to the task or were unclear, unspecific, or purely guesswork. For all tasks, the boys had 191 nonproductive steps coded as "identification" and the girls had 157.

The next most frequent nonproductive step was in planning (boys = 134, girls = 114). These were nonproductive steps regarding the input or resources necessary to complete the task. If the input or resources needed were either unrealistic, undetermined, unnecessary or incomplete, a nonproductive step in planning was recorded.

Finally, the least amount of nonproductive steps was found in the organization of the problem-solving procedure (boys = 77, girls = 79). These nonproductive steps were defined as steps which did not use the right order, sequence, or logic.

There were some instances in which the task was changed completely but processed very well. Therefore, a "changed task" category was created to keep track of these instances. Girls changed the task more frequently (13 times) than boys (8 times).

When analyzed by grade, the higher grades averaged less nonproductive steps than the lower grades (see Table 3). However the lowest grades had the best record for improvement. Third-graders had 3.9 nonproductive steps per attempt for Task I, and averaged 1.75 nonproductive steps per attempt for Task X, a reduction of 2.15. The fifth-graders reduced their average nonproductive step rate by 0.43. The eighth-graders actually showed a slight increase of 0.15 in nonproductive step rate from Task I to Task X.

Overall, the average number of nonproductive steps for all tasks when broken down by sex of subjects shows no appreciable difference (girls = 1.97, boys = 1.92). When the nonproductive steps are broken down by category, the boys averaged more nonproductive steps in identification and planning, and the girls averaged more nonproductive steps in organizing and changed the task more often. (See Table 3.)

The fifth-graders averaged more attempts or trials than the third- or eighth-graders. So, although completion rates for the fifth and eighth grades were the same, the fifth-graders needed more trials.

Before leaving the discussion of attempts at tasks, an incident occurred during pilot testing which alerted us to the existence of nonpersistent learners. A student who was very bright and seemed highly motivated and highly interested fled the room when faced with incomplete or incorrect results. In the field test, similar behavior was observed in every class, although the flight was as often

TABLE 3 □ Averge Number of Nonproductive Steps (all tasks by all attempts) for Grades 3, 5, and 8

	N	*Girls Average*	*N*	*Boys Average*	*Total Average*
GRADE 3	9	2.41	11	2.44	2.43
Nonproductive Steps	142		200		
Attempts	59		82		
GRADE 5	5	2.24	12	1.64	1.81
Nonproductive Steps	101		177		
Attempts	45		108		
GRADE 8	11	1.41	4	2.31	1.54
Nonproductive Steps	104		30		
Attempts	74		13		

psychological as it was physical. This phenomenon needs to be investigated further, since the modeling, structuring method of developing self-regulated learning had no effect on those students for whom learned helplessness and flight from failure has become their problem-solving model.

The mean completion rate for all three grades was 64%. The fifth and eighth grades had identical group completion rates of 73%, and the third grade had a 49% completion rate. Individually, some students in every grade completed all tasks, except Task IX, which was not assigned. There were also some students who requested and completed additional tasks. There was no appreciable difference in the completion rate of boys (63%) and girls (64%). The average number of attempts made for all tasks was 1.37 for all boys and 1.24 for all girls.

Finally, evidence of the development of self-regulation, i.e., learning to learn, was shown when those same students who had difficulty during week one teaching the computer to draw a square were able to direct the computer to draw cars, rockets, flags, clown faces, and a girl dressed in the 1950s style before the mini-course ended.

CONCLUSION

The process of learning to learn is not a simple one-way street or a trip which has been carefully mapped out by an auto club. In fact, the trip cannot be mapped by anyone other than the learner. At the same time, learners need not be left to wander without direction through mazes, detours, and wrong turns. Teaching students to utilize the metacognitive concept of self-regulation means that they will be able to identify goals, resources, methods, and monitoring strategies. It means they will get in touch with their own personal problem-solving skills, critically evaluate them, and then use them, revise them, or modify them as needed. Otherwise, the students will become like the traveller who Dewey (1933) says "in an unfamiliar region comes to a branching of the road. [He has] no sure knowledge to fall back upon, he is brought to a standstill of hesitation and suspense" (p. 13). Will he rely on luck to make his choice or can he discover "grounds for the conclusion that a given road is right?" (p. 13).

For the majority of students in this study, the model of teaching a computer to perform tasks provided access to personal problem-solving skills. The students were able to articulate "what" they knew and "how" they knew it. Once accessed consciously, those two necessary components of self-regulation—metaknowledge and metaprocessing—are open to utilization, change, or adaptation, depending on the goal.

This study did not specifically investigate the acquisition of new knowledge, processes, or strategies. However, the ability to access and engage both metaknowledge and metaprocessing and its articulation assures that the acquisition will be more meaningful and useful since the students will be able to supplement and modify their personal problem-solving strategies. This is the intent of self-regulated learning. □

REFERENCES

Brown, A. (1987). Metacognition, executive control, self-regulation and other more mysterious mechanisms. In F. E. Weinert & R. H. Kluwe (Eds.), *Metacognition, motivation and understanding* (pp. 65–116). Hillsdale, NJ: Lawrence Erlbaum Associates.

Brown, A. L., & DeLoache, J. S. (1978). Skills, plans, and self-regulation. In R. S. Siegler (Ed.), *Children's thinking: What develops?* Hillsdale, NJ: Lawrence Erlbaum Associates.

Dewey, J. (1933). *How we think*. Lexington, MA: D. C. Heath and Company.

Early, G. H. (1975). Learning disabilities: An information processing model. *The School Psychology Digest, 5*(1), 16–26.

Flavell, J. H. (1981). Cognitive monitoring. In W. P. Dickson (Ed.), *Children's oral communication skills*. New York: Academic Press.

Hartly, J. R. (1980). Computer assisted learning. In H. T. Smith & T. R. G. Green (Eds.), *Human interaction with computers* (pp. 129–159). New York: Academic Press.

Kelly, G. A. (1955). *The psychology of personal constructs*. New York: Norton Publishers.

Madinach, E., & Fisher, C. (1983). *Review of the research on the cognitive effects of computer assisted learning*. Berkley, CA: The Laurence Hall of Science, University of California and The Far West Laboratory for Educational Research and Development.

Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. New York: Basic Books.

Pea, R. D. (1985). Integrating human and computer intelligence. In E. L. Klein (Ed.), *New directions for child development*, (No. 28, pp. 75–96). San Francisco: Jossey-Bass.

Wellman, H. M. (1982). A child's theory of mind: The development of conceptions of cognition. In S. R. Yussen (Ed.), *The growth of reflections*. New York: Academic Press.

Statement of Ownership, Management and Circulation

1. Title of Publication: Educational Technology Research & Development, ISSN 1042-1629.
2. Date of Filing: 1989
3. Frequency of Issue: Quarterly. No. of Issues Published Annually: 4. Annual Subscription Price: $45.
4. Location of Known Office of Publication: 1126 Sixteenth Street, NW, Washington, D.C. 20036.
5. Location of the Headquarters or General Business Offices of the Publishers: 1126 Sixteenth Street, NW, Washington, D.C. 20036.
6. Names and Complete Addresses of Publisher, Editor, and Managing Editor: Publisher—AECT, 1126 Sixteenth Street, NW, Washington, D.C. 20036; Editors—Howard Sullivan and Norman Higgins, Arizona State University, Tempe, AZ 85287; Managing Editor—Blue Pencil Group, P.O. Box 3392, Reston, VA 22090.
7. Owner: Nonprofit Corporation: Association for Educational Communications and Technology, 1126 Sixteenth Street, NW, Washington, D.C. 20036.
8. Known Bondholders, Mortgages, and Other Security Holders Owning or Holding 1 Percent or more of Total Amount of Bonds, Mortgages, and Other Securities: None.
9. NA
10. Extent and Nature of Circulation.

	Average No. Copies Each Issue During Preceding 12 Months	Actual No. Copies of Single Issue Published Nearest to Filing Date
A. Total No. Copies Printed	0	4,000
B. Paid Circulation		
1. Sales Through Dealers, and Carriers, Street Vendors and Counter Sales	0	0
2. Mail Subscriptions	0	2,800
C. Total Paid Circulation	0	2,800
D. Free Distribution by Mail, Carrier or Other Means; Samples Complimentary, and Other Free Copies	0	1,000
E. Total Distribution	0	3,800
F. Copies Not Distributed		
1. Office Use, Left Over, Unaccounted, Spoiled After Printing	0	200
2. Returns from News Agents	0	0
G. Total	0	4,000

11. I certify that the statements made by me above are correct and complete. Stanley D. Zenor