DOCUMENT RESUME

ED 112 910 IR 002 621

AUTHOR Elliott, Portia C.; Peelle, Howard A.
TITLE Computer-Augmented Teacher Training.
PUB DATE Dec 74
NOTE 14p.; Paper presented at the Annual Meeting of the Association for Educational Data Systems (Virginia Beach, Virginia, April 29-May2, 1975)

EDRS PRICE MF-$0.76 HC-$1.58 Plus Postage
DESCRIPTORS *College Curriculum; Computer Assisted Instruction; *Computer Science Education; Mathematics Instruction; Program Descriptions; Programing; Student Teachers; *Teacher Education; Teacher Education Curriculum; *Teaching Methods; Teaching Techniques
IDENTIFIERS *Computer Augmented Teacher Training

ABSTRACT

Computer programing should not be confused with programed instruction (PI) or computer-assisted instruction (CAI). PI and CAI are processes that demand that the student assume a passive and docile role, whereas, programing is an active, dynamic process in which the student is required to instruct the computer how to perform certain basic operations. If the student is to give usable information to the computer, his decisions, planning, and thinking must be perfectly organized; thus, computer programing provides an excellent opportunity for the teacher trainee to acquire cognitive competencies in mathematics, to develop confidence with teacher methods, and to become aware of the benefits and limitations of computer technology. In order to acquaint future teachers with programing as a pedagogical tool, the University of Massachusetts has introduced a course entitled, "Computer-augmented Teacher Training." Among the expected goals of the course are to give the prospective teacher: (1) a knowledge of computers; (2) knowledge of algorithms and processes; (3) a knowledge of the applications of computer technology; and (4) experience in computer usage. (EMH)

COMPUTER - AUGMENTED TEACHER TRAINING

By

PORTIA C. ELLIOTT

And

HOWARD A. PEELLE

University of Massachusetts
Amherst, Massachusetts

December, 1974

For presentation at Association for Educational Data Systems Conference, April, 1975 (Based on article by same title to appear in Topics in Instructional Computing)

COMPUTER-AUGMENTED TEACHER TRAINING:

An Undergraduate Program in Mathematics Teacher Education

By

Portia C. Elliott and Howard A. Peelle
University of Massachusetts

INTRODUCTION

"Eventually, programming itself will become more important even than mathematics in early education."

--Marvin Minsky (13)

The preceding quotation suggests a dramatic departure from the proverbial three R's in elementary education and prophesies the inclusion of a powerful "P"--programming--in the curriculum of tomorrow. Programming is not to be confused with programmed instruction (PI) or narrowly construed uses of computer-assisted instruction (CAI) which place students in passive, almost docile roles by giving them prepackaged programs with preordained answers and prescribed paths to those answers. Rather, programming (in an education context) involves students actively directing the computer to solve problems or to create original projects. Programming is fundamentally a learner-centered activity--one which allows the learner to take increasing responsibility for his own educational destiny.

Recently, other computer scientists and educators have envisioned programming in the elementary school classroom. Dwyer (4), Luehrmann (11), Milner (12), Papert (14), and Peelle (16), all have advocated student-

controlled computing and non-exploitative uses of computer technology in education. Papert's (14 (p. 1)) view of an ideal relationship between student and computer is particularly refreshing:

> . . . technology is used not in the form of machines for processing children, but as something the child himself will learn to manipulate, to extend, to apply to projects, thereby gaining a greater and more articulate mastery of the world; a sense of power or applied knowledge and a self-confidently realistic image of himself as an intellectual agent.

Many of the activities and concepts involved in programming a computer are, indeed, important to learn. The notions of "bug" and "debugging;" of algorithms (procedures) and sub-procedures; of iteration and recursion; and of feedback and heuristics are intellectually "powerful ideas" (to borrow again from Papert). They are ideas we rely upon every day--perhaps unconsciously--to process information from the world we perceive and to make decisions, plan, think, etc. In brief, these ideas are important tools which can serve as building blocks for further cognitive development.[1]

The question then arises: 'What about teachers and their development?" What knowledge and what skills should teachers have when computer programming is accepted in elementary schools? The answer is not found in the simplistic suggestion: "Take a computer course." Most (so-called) computer courses are not designed with applications for teaching in mind. And yet, if concepts of programming are valuable cognitive tools for students to have, are they not for their teachers as well?

---

[1]Some preliminary research efforts have attempted to connect Piaget's developmental theories to the learning of programming concepts. See dissertations by Michael Folk, "Influences of Developmental Level on a Child's Ability to Learn Concepts of Computer Programming," (Syracuse University), 1972; and Joyce Statz, "The Development of Computer Programming Concepts and Problem-Solving Abilities," 1973, (Syracuse University).

Unfortunately, teacher training in computer usage has been mostly neglected. In their report, "Factors Inhibiting the Uses of Computers in Education," Anastasio and Morgan (1) cited lack of adequate teacher training programs as a major deterrent to computer use in schools. The Conference Board of the Mathematical Sciences' report (2) listed the following as one of their recommendations:

> We recommend. . . the development of a variety of programs for the training of teachers and of teachers of teachers of high school courses involving computers.

But, teachers themselves are notorious for resisting innovation. They seem to harbor a lot of myths and fears about technology--particularly about computers. For some teachers, the computer is seen as a vulgar intrusion into their private domain--the classroom--and threatens their very jobs; for others, computers are wonder-machines which "can do anything" or (worse yet) are "smarter than people;" and for many, there is a deep reluctance to use sophisticated equipment--usually expressed as "No thank you. . . I might break something" in order to avoid exhibiting ignorance. These fears and myths are, of course, largely based on misunderstandings or lack of understanding about computers.[2]

So, the problem for teache education remains: How should we train teachers who will be exposed to computers and programming? Granted, not all teachers can teach programming effectively; some may become expert in other aspects of computer usage, such as curriculum development. But at a minimum, teachers need to be able to convey the importance of knowing about computers and be able to arrange computing environments which foster the development of competent learners.

---

[2]For further explication of this point, see Bork (3) and Peelle (15).

COMPUTER-AUGMENTED TEACHER TRAINING -- AN OVERVIEW

A new teacher education program has been inaugurated at the University of Massachusetts' School of Education under the title: "Computer-Augmented Teacher Training" (abbreviated CATT). CATT is an undergraduate program designed to train prospective teachers of mathematics.[3]

The primary goal of the CATT program is to develop teachers' competencies, utilizing the computer where and when it augments their training. Three sub-goals of teaching competency are identified as follows:

1. Developing Cognitive Competencies

   Development of competence in mathematics content areas (what to teach); and development of competence in methods of effective teaching (how to teach).

2. Developing Affective Competencies

   Development of self-concept (confidence in teaching) expression and management of emotions (self-control).

3. Developing Social Consciousness

   Learning the advantages and limitations of technology (what computers can and cannot do); understanding man-machine interaction (computer literacy); awareness of critical issues in a technological society.

---

[3] While this program is open to prospective teachers of mathematics at all levels--elementary, intermediate, and secondary--it is described here only in terms of elementary mathematics teacher training. (A complete description may be obtained from the authors.)

COGNITIVE COMPETENCE: KNOWING WHAT TO TEACH AND HOW TO TEACH

In order to strengthen teachers' cognitive competencies, the CATT program offers training in both content and methodology; that is, prospective teachers are taught what to teach and how to teach. Trainees develop their understanding of mathematics as they are re-introduced to fundamental mathematical concepts via a programming language. (See "Learning Mathematics Via A Programming Language" below.) Trainees also become acquainted with pedagogical strategies for teaching mathematics effectively, emphasizing use of computer technology and programming paradigms. (See "Teaching Mathematics Via A Programming Language" below.)

Learning Mathematics Via A Programming Language

Based on the idea that "a good way to learn something is to teach it," teachers-in-training in the CATT program "teach" the computer. That is, they learn mathematics by communicating their newly assimilated knowledge to a computer, using A Programming Language.[4]

The role of the computer is like that of a "model student." The computer responds only to explicit instructions; it does exactly what it is told (even if that is different from what one though one told it); once it is "taught" something (like a rule), it does not forget or distort it and can apply it accurately upon command; and it is (usually) ready to accept more instruction, tirelessly and obediently.[5]

---

[4]APL (A Programming Language) is a general-purpose interactive computer programming language developed by Kenneth Iverson (7) and supported by IBM. Originally conceived as a unifying mathematical notation, APL is a language with simple rules and yet offers a user great computational power and flexibility. APL has been applied successfully in fields such as business, scientific research, and education. (See Iverson (8))

[5]If a computer is not available, the activity of "dry" programming is still valuable for cognitive development. See Iverson (9) for a description of using A Programming Language as a notation for clarifying concepts and procedures.

Thomas Kurtz (10) and the Dartmouth Secondary School Project staff are proponents of this "computer-as-pupil" approach for high school students:

> Because we had to teach an ignorant machine, we were forced to break the process down into pieces, arrange these pieces in the proper order, and present them to our pupil machine and see if our instructions were presented in a logical, foolproof way that it could follow. Before we made the effort to teach this pupil, we were forced to clearly understand the problem ourselves.

The approach applies equally well to teacher training.

Teaching Mathematics Via A Programming Language

In this aspect of their training, prospective teachers become acquainted with strategies for teaching mathematics effectively. This includes the pedagogy of using the computer directly in the teaching process. Not only do these teachers learn about computer use and its application in mathematics, but also they are shown how computer programming, in particular, can be used as a means for teaching mathematics.

Why use computer programming in training teachers? Because certain processes intrinsic to programming a computer are analogous to some powerful methods of teaching. For instance, only by programming can prospective teachers learn to control the computer and "debug" programs (eliminate errors). Just as a teacher must learn how to handle academic and disciplinary "bugs" in the classroom, s/he can find strategies which work for "debugging" computer programs.

In the role of "model student" (described earlier), the computer allows one to do some "practice teaching"--with no detriment to real students! The teacher treats the computer as a simulated student--with potential for developing its innate abilities--and may observe the effects of different teaching approaches.

A variety of pedagogical strategies are demonstrated to the prospective teacher. The strategies are actually a set of heuristics which teachers may use to facilitate student learning activities.[6] For example, they might:

--Use Manipulative Materials

--Encourage Articulation of Thinking Processes

--Encourage Articulation of Emotional Reactions

--Allow Increasing Learner Control

--Change Pedagogical Strategies

AFFECTIVE COMPETENCE: SELF-CONTROL AND CONFIDENCE IN TEACHING

In order to develop the affective competencies of teachers, they are placed in active learning environments--environments in which both success and failures are likely to occur. (This is characteristic of the type of environment in which they will be eventually teaching.) That is, by actively dealing with their own emotional reactions to success and failure, prospective teachers can improve their self-control and develop confidence in their teaching ability.

Once again, the programming paradigm helps. When a teacher-in-training writes a program, it either works properly or it doesn't. Success or failure is easily recognized. By contrast, in huma affairs, success and failure are often ambiguous and difficult to identify. When

---

[6]These are "heuristics" in the sense that they are not guaranteed to work for all students at all times. See Dwyer (4) for a broader description of "Heuristics for Using Computers to Enrich Education."

a program is the object of attention, there can be joy in seeing that "it works!"; or, of course, there can be frustration caused by irrepressible "bugs". But, in either case, an emotional reaction is illicited; and that reaction is explicit and focussed--but not at another person--making it easier to deal with.

Learning to manage one's emotional reactions can be facilitated by the presence of a computer. When a programmer is successful, s/he has a medium available for automatically and proudly displaying results. When a programmer is not successful, s/he can debug the program in private consultation with the computer and work through the accompanying emotional stress without any additional social embarrassment. The motivation for dealing with failure constructively is strong--for things will certainly not get any better unless you do something about it. After all, the computer cannot fix your program!

SOCIAL CONSCIOUSNESS: COMPUTER LITERACY

In order for teachers to become more aware of the extensive commitment our society has made to technology, the CATT program is designed to develop their social consciousness. Cognitive and affective teaching competencies are not enough. Teachers need to know about the advantages and limitations of advanced technology--in particular, what computers can and cannot do.

Developing such a "computer literacy" is important if a teacher is to effectively deal with problems of a technologically-oriented society. Several studies of computer use in education have cited computer literacy

as a high priority.[7] Accordingly, the CATT program prepares prospective teachers with special skills, such as ability to:

--identify cases when technology is used as a shield behind which decision-makers hide, e.g., "The computer gave us this answer" (school busing route);

--critically analyze curriculum materials for negative racial or sexist overtones, particularly computer-based curriculum;

--debunk myths (e.g., computer omnipotence) and assuage fears (e.g., "The computers are after our jobs").

These skills translate back to computer literacy. For, as Alvin Tofler (17) put it: "Tomorrow's schools must. . . teach not merely data, but ways to manipulate it." The computer is man's most powerful and versatile tool for manipulating data. Indeed, it is a supreme symbol manipulator--performing instructions at speeds measured in nanoseconds, which far exceed man's own capabilities.[8]

But, the major reason for including computer literacy as a significant component of a teacher education program is not so that teachers (or anyone else) will compete with computers. Rather, teachers should be at the forefront, charged with the responsibility for teaching computer literacy (along with other essential literacies). They themselves must, therefore, learn how and why computers are used. Specifically, this involves:

---

[7] See, for example, Begle (2).

[8] In the area of pattern recognition, however, the computer does not fare as well, comparatively. A major area of research in artificial intelligence is devoted to what humans regard as relatively simple tasks of pattern recognition, such as distinguishing between (script) letters of the alphabet.

--Knowledge of computers (what a computer is; how a computer works; etc.)

--Knowledge of computing (information processing; algorithms, etc.)

--Knowledge of applications (what computers can do/can't do)

--Experience in computer usage (man-machine interaction)

--Using computers in teaching (computer-assisted instruction; computer managed instruction)

--Programming (how to control the computer).

--Issues and Implications of Computer Ubiquity (data banks vs. privacy; artificial intelligence, etc.)

CONCLUSION

In summary, the CATT program described here has several distinguishing features which make it unique in teacher training. First, the computer --and A Programming Language in particular--play a central role in the development of prospective mathematics teachers. Second, modern technology is interwoven with traditional teaching approaches and content is blended with methods to make a palitable mixture for teacher trainees. Finally, the program stresses development of cognitive, affective and social competencies in prospective teachers to ensure that future learners will have teachers who are more knowledgeable, more sensitive, and more humane.

REFERENCES

(1) Anastasio, E. and Morgan, J. National Science Foundation Report: "Factors Inhibiting the Uses of Computers in Education," 1972.

(2) Begle, Edward et al. Report of the Conference Board of the Mathematical Sciences Committee on Computer Education: "Recommendations Regarding Computers in High School Education," April, 1972.

(3) Bork, Alfred. "The Computer in Teaching--Ten Widely Believed Myths," ACM SIGCUE Bulletin, Volume 7, Number 4, October, 1973.

(4) Dwyer, Thomas A. "Heuristic Strategies for Using Computers to Enhance Education," International Journal of Man-Machine Studies, 1974.

(5) Elliott, Portia C. "Elementary Mathematics Teacher Training Via A Programming Language." Unpublished Ed.D. dissertation, University of Massachusetts, 1973.

(6) Furth, Hans and Wachs, Harry. Piaget's Theory in Practice: Thinking Goes to School. New York: Oxford University Press, 1974.

(7) Iverson, Kenneth E. A Programming Language. New York: John Wiley and Sons, Inc., 1962.

(8) Iverson, Kenneth E. "APL in Exposition," IBM Technical Report, Number 320-3010, Philadelphia Scientific Center, January, 1972.

(9) Iverson, Kenneth E. "APL As An Analytic Notation," Proceedings of APL V Conference, Toronto, Canada, May, 1973.

(10) Kurtz, Thomas E. "The Computer As Pupil: The Dartmouth Secondary School Project," NSF GW-2246, Final Report, October, 1970.

(11) Luehrmann, Arthur W. "Should the Computer Teach the Student or Vice-Versa?", AFIPS Conference Proceedings, Volume 40.

(12) Milner, Stuart. "Learner-Controlled Computing: A Description and Rationale," Paper presented at the Annual Meeting of the American Educational Research Association, Chicago, Illinois, April, 1974.

(13) Minsky, Marvin. "Form and Content in Computer Science," Journal of the ACM, Volume 17, Number 2.

(14) Papert, Seymour. "Teaching Children Thinking," MIT Artificial Intelligence Memorandum Number 247.

(15) Peelle, Howard A. "Pygmalion's Computer," in Controversies in Education, edited by Allen and Hecht. Saunders, Inc., 1974.

(16) Peelle, Howard A. "COMPUTER GLASS BOXES: Teaching Children Concepts With A Programming Language," Educational Technology, Volume XIV, Number 4, April, 1974.

(17) Tofler, Alvin. Future Shock. New York: Random House, 1970.