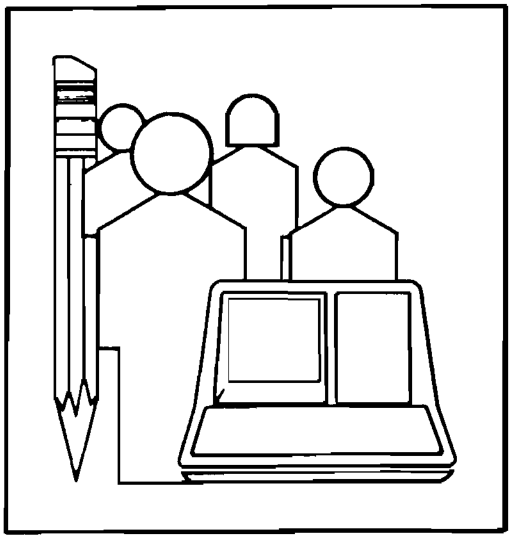

TECHNICAL REPORT

NEW BIRD ON THE BRANCH: ARTIFICIAL INTELLIGENCE
AND COMPUTER-ASSISTED INSTRUCTION

Technical Report No. 16

August 1983

DIGITAL EQUIPMENT CORPORATION
EDUCATIONAL SERVICES
12 Crosby Drive
Bedford, Massachusetts 01730

digital

NEW BIRD ON THE BRANCH: ARTIFICIAL INTELLIGENCE AND COMPUTER-ASSISTED INSTRUCTION

Jo-Anne Wyer

ABSTRACT

When Artificial Intelligence (AI) "marries" Computer-Assisted Instruction (CAI), the result is "Intelligent CAI" (ICAI). This paper surveys some of the intelligent CAI programs that have been authored, and differentiates them from traditional "frame-oriented" CAI.

ICAI programs have several characteristics in common. They differentiate the material to be taught from the method of teaching; they allow students to learn via discovery and develop their problem-solving skills; they create an internal "model" of the student's learning; they exhibit a variety of tutoring strategies. ICAI programs attempt to emulate the best traits of human tutors and have done so with varying degrees of success. While the frame-oriented style of CAI seems to be a cost-efficient information delivery system, the ICAI approach perhaps uses the computer to its fullest potential while at the same time being more responsive to the student.

There are many contraints, however, on adopting ICAI in a wide-spread manner. ICAI systems are complex and exacting to author; the use of natural language with computers is still in its infancy; computing power is often at a premium. Nevertheless, it seems worthwhile to entertain the possibilities for computers in teaching, not necessarily as a substitute for any other (existing) media, but for the unique contribution this virtually untapped resource can make.

This paper was written as a term paper for a "Seminar on Computer-Assisted Instruction" taught at Boston University by Dr. Lisa Ehrlich.

INTRODUCTION

Science fiction writers have been promising us intelligent machines for decades. How close are we to realizing that notion? And, what benefit will it have for education, or computer-assisted instruction in particular?

The pursuit of this kind of question comes from an analysis of the predominating methodology of existing instruction which is computer-based. To the unsophisticated inquiring eye, it appears that most CAI takes little advantage of the tremendous computing power available even in today's microcomputers. Simple, repetitive drill-and-practice; highly controlled, predictable branching; in short, programmed instruction on a video monitor. Useful, certainly, for certain kinds of instructional intent, with the additional features of being able to monitor and record student progress, but somehow appearing to sprout helter-skelter on an untilled -- and presumably untried -- fertile field.

There seem to be two facets to the question of how research in the field of artificial intelligence can influence the field of computer-assisted instruction. The first is: what is artificial intelligence, and how can it be implemented to expand the usefulness of CAI? The second facet reflects a more pragmatic intent: what benefit will these types of CAI have for us and how can we use them? Will they, in fact, become practical teaching methods within school systems as we currently know them, higher education, or industry?

WHAT IS ARTIFICIAL INTELLIGENCE?

M. L. Minsky, a specialist in the field, defines artificial intelligence as "the science of making machines do things that would require intelligence if done by men" (as quoted in Boden, 1977). This definition itself requires some analysis, for it never implies that machines themselves will have intelligence (the ability to learn, reason, or understand), but only that they can be made to exhibit behavior which appears to be guided by intelligence. The distinction is an important one, because it points directly to the kinds of research being done in the AI field. The objective is not to create a machine that can think, but rather to use machines to model thinking. The key here is not hardware, but the attempt to define software systems which effectively imitate human reasoning. These efforts have had some startling results.

The program is the soul of the computer. The program codifies meaning into representational symbols. The computer manipulates these symbols at high speed and produces output. Again, it is the person who adds the layer of meaning to the outputted symbols. Since meaning is separate from how the symbols are manipulated, there is enormous flexibility in computer programming. One can use the constancy of Boolean logic to approximate the contextual, often para-logical or quasi-logical, behaviors of human beings.

K. M. Colby, a practicing psychoanalyst, developed a program which simulates the interchange between a psychotherapist and a patient, a neurotic woman who believes her father abandoned her but is is unable consciously to accept the fact that she hates him. (See reference in Boden, 1977.) Colby attempted to develop a theoretical model of the psychoanalytic therapeutic process for this particular patient. By feeding in all the variables gleaned from hundreds of hours of therapy, he hoped to be able to "test out" therapeutic strategies. Such a program, Colby suggested, might also be used for training.

While such a program seems wildly "intelligent" on the surface, it is an early program (1960's). The actual dialogue between therapist and patient is primitive and almost dissociated. For example:

Program: FATHER IS INDIFFERENT TO ME.
MOTHER SAYS FATHER ABANDONED US.
FATHER PREFERRED SISTER.
I AVOID FATHER.
SISTER ADMIRED FATHER.
I FEEL TENSE NOW.

Therapist: YOU ADMIRE FATHER

Program: NO.
I DO NOT ADMIRE FATHER.
I ADMIRE AL.
MOTHER OUGHT NOT TO ADMIRE AL.
MOTHER DETESTS FATHER.

The reason for this primitive interchange, briefly, is that Colby's data structure is not as sophisticated as any person's knowledge structure. In programming, he has set up relationships, dependencies, "things," i.e., emotions, thoughts, meanings, which cancel each other out or call each other into being. He has attempted to mimic meaning, but the result is still blatantly artificial.

What is important, however, is the attempt to build that data structure. Margaret Boden, in _Artificial Intelligence and Natural Man_ (1977), perhaps puts it best:

> Knowledge ... is embodied in a program in two ways: in the memory, or data base, and in the procedures that operate on or by reference to the data. One of the strengths of

artificial intelligence as a way of thinking about thinking is that it forces one to consider the dynamic aspects of intelligence. A functioning program is a theory that is intended more as a movie of the mind than as a portrait of it, and the programmer must specify precisely how successive frames are brought into being. Many verbal theories, like those of Freud himself, likewise are attempts to model the movement of thought. But it is all too easy, when theorizing in verbal terms, to imagine that one has made matters explicit that in fact one has not. Since only a program can tell a computer to do, the programmer's largely intuitive psychological theory must be expressed in computational terms. It is not enough to say that a certain progression of thoughts can happen: the program must represent how it can happen.

Movies of the Mind

The attempt to chart the movements of thought characterizes most research in the artificial intelligence area. Generally, researchers have looked at what the human mind can do and attempted to create its computational equivalent. Research efforts have generally fallen into three broad categories:

- Pattern Recognition and Image Processing
- Natural Language Parsing
- Knowledge-Based Systems

Research in the areas of pattern recognition and image processing have largely dealt with the issues of extrapolation from the simple, or easily discernable, to the complex, or obtuse. Work has been done to program computers to "recognize" and often reproduce graphically shapes which have been stored in their memory as abstractions. In the interest of brevity, suffice it to say that such experiments have pointed to the human mind's remarkable abstracting powers. But how can a computer program be taught to see? Must it always be "presciently provided with the representational theories they will use to interpret the world?" If so, the effort required to give a computer a brain seems formidable. Even so simple a task as we perform every day -- distinguishing one person's face from another -- becomes a programming nightmare.

ELIZA is a fairly famous example of a program masquerading as a person -- and of research in the area of natural language parsing. (See discussion in Boden, 1977.) ELIZA is a non-directive therapist; the user is the patient. A typical therapy session might go like this:

Patient: Men are all alike.
ELIZA: IN WHAT WAY
Patient: They're always bugging me about something or

	other.
ELIZA:	CAN YOU THINK OF A SPECIFIC EXAMPLE
Patient:	Well, my boyfriend made me come here.
ELIZA:	YOUR BOYFRIEND MADE YOU COME HERE
Patient:	He says I'm depressed much of the time.
ELIZA:	I AM SORRY TO HEAR THAT YOU ARE DEPRESSED
Patient:	It's true. I'm unhappy.
ELIZA:	DO YOU THINK COMING HERE WILL HELP YOU NOT TO BE UNHAPPY
Patient:	I need some help, that much seems certain.
ELIZA:	WHAT WOULD IT MEAN TO YOU IF YOU GOT SOME HELP
Patient:	Perhaps I could learn to get along with my mother.
ELIZA:	TELL ME MORE ABOUT YOUR FAMILY ...

What ELIZA demonstrates and what could be construed as intelligence is really a clever pattern matching program that depends heavily on the unique characteristics of a non-directive counselor.

Taken out of context, this type of conversation might seem inane, but in the context of therapy, it is acceptable and, in fact, quite effective. Weizenbaum's work with ELIZA is a good example of technique in natural language parsing. Here, responses are scanned for keywords, and the sentence transformed according to the rule associated with the matched keyword. If no keyword is found in the input text, ELIZA responds with either content-free formula (WHY DO YOU THINK THAT?) or a reference to some earlier remark (DOES THAT HAVE ANYTHING TO DO WITH THE FACT THAT YOUR BOYFRIEND MADE YOU COME HERE?). Substitution rules make sure that the pronouns are always appropriate ("I" versus "YOU"). Something similar to playing a conceptual game of billiards.

A more advanced program exhibiting natural language "understanding" is represented in a robot named SHRDLU. Basically, SHRDLU can follow directions, sort out meanings, and tell a big red block from a little blue one. A typical user/program dialogue might look like this:

User:	Pick up a big red block.
SHRDLU:	OK.
User:	Grasp the pyramid.
SHRDLU:	I DON'T UNDERSTAND WHAT PYRAMID YOU MEAN.
User:	Find the block which is taller than the one you are holding and put it into the box.
SHRDLU:	BY "IT", I ASSUME YOU MEAN THE BLOCK WHICH IS TALLER THAN THE ONE I AM HOLDING.

The difference between ELIZA and SHRDLU is that the latter has been given a system of knowledge. The program can draw on that system of knowledge to make "reasoned" judgements and to query you. The only flaw might be that the program's knowledge base is quite limited, the universe being comprised of much more than tall red boxes (including pyramids), but the quality of the language interaction between man and machine is much improved, and offers much promise.

Knowledge-Based Structures

Several AI programs which we've talked about already demonstrate some sort of underlying assumptions about the structure of a particular domain of knowledge. We've already touched on how the structure itself is one concern; the conceptualization and programming of interrelationships quite another. But the fundamental idea is that certain software systems set out to "mimic" the deductive and inductive reasoning of the human mind. The result has been an effort to set up "networks" of interrelationships.

Machines Imitate Life

Jesse Heines (1983) has outlined how knowledge-based systems are structured. Basically, the data base (or information base) is programmed to have within certain logic rules by which the program operates, and which control branching. Interaction by the user/student is analyzed to determine which rules apply, and the result is in the form of a change in the inference network accompanied by some sort of feedback to the user. The logic constructs, he goes on to explain, are based not on the strict Boolean logic which is the basis for hardware (If A, then B), but on Bayesian logic which allows for the intricacies and interdependencies which underly most human decisions (If A to degree X, then B to degree Y).

It is in the area of knowledge structures that the fields of artificial intelligence, cognitive psychology, and computer-assisted instruction come together, for in all three there is the central core issue: how do we learn? It is only with a better understanding of how human beings assimilate and structure knowledge that we can build better teaching aids or more intelligent machines.

INTELLIGENT COMPUTER-ASSISTED INSTRUCTION

A brief history of CAI will help to position the potential of artificial intelligence on this field.

CAI has its origins in behavioral psychology. Sidney Pressey's teaching frames required the mastery of the first before going on to the second, etc. (Pressey, 1926, 1927). Norman Crowder's "scrambled textbooks" introduced the idea of branching dependent upon student's responses (Crowder, 1960). A third level, called "adaptive," based the branching not on single responses, but on a history of responses based on building a rather sketchy model of the student which was stored between sessions (Gable and Page, 1980).

What all of these models and much current-day CAI have in common is the environment of controlled learning. This type of learning environment might be called "Ad Hoc Frame-Oriented CAI" (AHFO CAI). From an artificial intelligence perspective, AHFO CAI has the following limitations:

- The student can take little or no initiative in his own learning.
- The student can't use natural language with the system.
- The systems look fairly rigid to the student, perhaps heightening frustration. (Clancy et al., 1979.)

"Intelligent" CAI (ICAI), as it has been called, attempts to approach the learning situation from a less controlled, less technology-driven and more student-driven perspective (see Sleeman and Brown, 1982). The underlying assumption is less that the learner is a passive respondent, and more that s/he is actively involved in generating his or her own knowledge base. As such, it takes a turn from CAI's behavioral inheritance, and borrows more from the field of cognitive psychology.

Generative CAI

Turning from AHFO CAI, where all presentation of information and branching was fixed by the author, "generative" CAI was introduced (see discussion in Gable and Page, 1980). Generative CAI stresses the ability to generate problems for the student to solve using a large stored database representing the subject matter being taught. All decisions are made dynamically, on the basis of student interaction, as the student goes through the material. Early generative programs were mathematical proof checkers, where students used the system as a critic for their mathematical proofs. But a true departure was Carbonell's SCHOLAR program (1970). SCHOLAR departed from the AHFO approach in that its structure was information oriented, designed to teach the geography of South America through a "mixed-initiative" system rather than through frame-oriented "page turning". In other words, the student could query the system, using natural language, and vice versa.

SCHOLAR's comprehension and generation of natural speech is based on a semantic network which is in turn based on studies of natural language comprehension by Quillian and others (see Boden, 1977). The content of this network is built by the teacher feeding the information into the system, which codes it according to the structure of the network. The result is that the system can ask a question like: "What is Chile?," and the student can answer, "A country." SCHOLAR will understand this, and the system will output, "Correct." The student can also input, "Please tell me more about Peru," to which SCHOLAR can reply, "The area of Peru is approximately 480000 square miles; the language is Spanish;" etc., to the extent that the SCHOLAR has been programmed to have information stored under the code: "Peru."

SCHOLAR was significant for ICAI in that it made some attempt to utilize restricted natural language for both input and output, but it did not meet another requirement set forward for intelligent CAI systems: it makes no attempt to evaluate student's incorrect responses or to use a diagnosis of what may be wrong to help the student overcome misunderstandings or misconceptions. This touches on one of the drawbacks of much frame-oriented CAI, and is perhaps indicative of a certain narrowness of perspective that behavioral-based teaching methods can have; that is, an overriding concern for successful outcome while ignoring the enormous issue of how one gets there.

Proponents of ICAI would argue that AHFO CAI not only represents (however inexplicit or unintended) the theoretical assumption that knowledge is and can be subdivided into lessons which are "fed" to the student by the teacher on a frame-by-frame basis, but also a poor utilization of the real powers of a computer. Well, all economic and pragmatic constraints aside -- and there are many in the real world -- we'll briefly review some of the more intelligent CAI systems.

Teacher, Tutor, Expert, Coach

Much thinking in the area of intelligent CAI is based upon the principle that teaching strategies must be separated from the subject matter to be taught. When the two are separate, there will more flexibility in how the student can interact with the data. Theoretically true, but in implementation very difficult.

> ... the separation of subject-area knowledge from instructional planning requires a structure for organizing the expertise that captures the difficulty of various problems and the interrelationships of course material. Modeling a student's understanding of a subject is closely related conceptually to figuring out a representation for the subject itself or for the language used to discuss it. (Clancy et at., 1979.)

Designing any courseware or teaching unit of course presumes that we are, at lease intuitively, supposing the optimum methods for understanding it. When we, as course designers, or teachers, can control that presentation to the student, we feel more secure of the outcome. It is also a simpler design problem. To give the learner more control of his learning environment triggers not only some insecurity on our part, but also more questions about how that environment should be constructed and, ultimately, is there more than one way to learn?

PLATO is probably one of the most famous CAI systems which allows some learner control. The system sets up problems or questions for the student to answer. The student thinks about what information s/he needs, interprets the data gathered, and tests assumptions. The system provides feedback on those assumptions. While remarkable in other respects, PLATO does not work with the student feedback to build a model of the student on which to base future interactions. PLATO is more a black box "expert" than a tutor.

In effect, most intelligent CAI systems have attempted to profile successful tutors. Because the presentation of information is more learner-driven, the focus of the systems has been to guide the student's learning through interactivity. SOPHIE is an impressive example. SOPHIE is intended to help teach electronic engineering students how to locate the faults in malfunctioning electrical circuits. The student is presented with a circuit fault of some specified difficulty. The student can then request more information of the system, like a schematic diagram or measurements under any instrument settings. When the student feels ready, s/he can generate a hypothesis which is checked by the system. More importantly, if the student is stuck, s/he can ask for help and SOPHIE will generate possible hypotheses using the student's measurements. The most exciting aspect of SOPHIE is that its problem-solving environment is so easily transferable to reality; it is game-like with useful feedback and plays so well into the actual logical deductive strategy the student himself or herself would use on the job.

Stanford University's Basic Instructional Program (BIP) is an attempt to use the computer as a tutorial laboratory for teaching programming in the BASIC language. Based on the assumption that computer programming, like other kinds of procedural knowledge, is best learned by doing, BIP presents the student with sets of programming "problems." Based on the student's performance on these problems, which have a prescribed range of difficulty, BIP can begin to make a model of the student's state of knowledge. As such, it takes an important step: that between just recording the student's history and selecting the next problem for him or her. It basically assess the student's skill set and assigns problems to improve those skill sets. Thus students may get the same problems, but for different reasons.

BIP is based on a Curriculum Information Network which is deliberately and explicitly devised, but which is not presented to the learner frame-by-frame. In such a way, it hopes to represent the best of both worlds:

> It allows meaningful modelling of the student's progress along the lines of his developing skills, not just his history of right or wrong responses, without sacrificing the motivational advantages of human organization of the curriculum material. (Barr et al., 1976.)

It is also, perhaps, one of the only examples of true individualization of learning.

One of the distinctive characteristics of ICAI is student modelling. This is one of a human teacher's strongest tools, and ICAI attempts to develop student models to effect a better match between the instruction and the student. BUGGY (Burton and Brown, 1979) is a computer-based tutoring/gaming system devised to teach students how to diagnose "bugs" in their mathematical reasoning and thus to attain a better understanding of the underlying structure of mathematical skills. Students are presented with a series of mathematical problems to solve. When problems are solved incorrectly, the program attempts to discover the "bug" in the student's thinking and to offer appropriate tutorial advice.

One of the interesting aspects of this program is almost an aside. The authors based this program on a certain hypothesis, namely that teachers often think that students are bad procedure followers. The authors thought the opposite: that pupils follow procedures very well and that mistakes are often the result of following the wrong procedures. Interestingly, students who tried BUGGY in the classroom progressed from a disdain for the "stupidity" of the program to an appreciation that there was, in fact, "a systematic explanation for what the program was doing."

WEST is also an intelligent game which attempts to diagnose student problems unobtrusively and offer help. As such, it is termed a "coaching" system because it allows students to make decisions freely and observe the results; the coaching is not intrusive. WEST is principally a computer board game where students use both luck and gaming strategy to win. The game involves not only knowledge of basic arithmetic, but game playing strategies as well and the coach is called in to play to help the student with both. The computer coach is the "expert," but the overt objective is to "win" not to learn. Yet, interestingly, students who are "coached" seem to enjoy the game better!

The last system that we'll talk about is GUIDON, a "case method tutor" who uses the information base of MYCIN to tutor students in medical problem-solving. MYCIN was originally designed as a diagnostic program or, in effect, a passive teacher. MYCIN would query a physician about symptoms of a patient until it had enough information to output recommendations about diagnosis and therapy. GUIDON adapted that knowledge base to a tutorial where the student plays the role of physician consultant. The program compares the student's responses to MYCIN's internal rules and critiques him or her. The significance of GUIDON is that the system pursues a more exacting course of instruction in that the dialogue with the program is managed according to some specific rules. While the coaching programs takes a very unobtrusive approach, GUIDON adopts different dialogue formats according to preset rules. For example, the kind of dialogue pursued when introducing a new topic will be different from that used when a student asks a question demonstrating some unexpected expertise.

ISSUES IN INTELLIGENT CAI

ICAI seems to pursue some strategies which its brother CAI does not. It attempts to instill problem-solving expertise, it attempts to "model" the student, and it devises particular tutoring strategies. All three of these components seem to point to a utilization of the computer for things it is good at: providing models of reality that can be manipulated (problem-solving); tracking and storing information (student modeling); and defaulting to many varieties of programmed paths (tutoring strategies). In theory, the implementation of these types of teaching programs should suit computers and their programmers ideally -- and CAI would take a giant leap ahead as an educational tool.

But, there are problems. There are problems with resources, with economics, with computing power availability, etc. But more than that, ICAI is still as yet a seedling. The field as yet appears to be small, and successful projects are few in number. The ICAI development process itself is full of unanswered questions. It is relatively manageable to undertake design of a learning package which imparts information to a student, whether frame-by-frame or via visual media or lecture. It is far more difficult to design access to an information structure which is user-controlled. In addition, it is quite another matter to model the acquisition of skills and to design programs which allow for all possibilities of reasoning or misunderstanding. In short, while the benefits of truly individualized instruction seem to expand exponentially, so do the design difficulties. The questions then become: what would it take to accomplish more in ICAI, and is it worth it?

FINAL REMARKS

> "And how many hours a day did you do lessons?" said Alice, in a hurry to change the subject.
>
> "Ten hours the first day," said the Mock

> Turtle, "nine the next day, and so on."
> "What a curious plan!" exclaimed Alice.
> "That's the reason they're called lessons," the Gryphon remarked: "because they lessen from day to day."
>
> Lewis Carroll
> Alice in Wonderland

Of course, we have been talking about revolutions in education longer than we've been talking about the evolution of intelligent machines, yet the arrival of machines that learn seems more certain than a revamping of the educational system we so often find fault with. Many of the issues involved go well beyond the scope of this paper. They are psychological, sociological, political, economic, even philosophical. They guide the way we think about how people should learn and cloud our investigation of how people do learn. They perhaps limit our creativity, as we often limit the creativity of those we teach, by focusing on limitations and not possibilities. By striving painfully to avoid making an error, we suffer a loss of freedom. By adopting this as our working theorem in teaching, we may encourage the status quo and punish the visionaries.

How does this relate to CAI? By taking a strictly frame-oriented, passive-learner, right-or-wrong answer approach, we are, in fact, espousing certain assumptions about how people learn, what we will allow them to learn, and how they should be taught. This may, in fact, be a valid strategy in certain situations and under certain constraints -- but it should be recognized as such. It is not important, I believe, that we have our Theory of Learning fully spelled out before we proceed, but only that we recognize that any presumption to teach assumes that we have one. If we adopt a certain medium and/or method, it should be based upon those assumptions unless prohibited by constraints beyond our control. On the other hand, to "stretch" CAI -- to attempt to adopt AI techniques to "live" courseware requires a researcher's budget and schedule.

So is it worth it? Probably yes. The computer's potential for accomplishing truly worthwhile things is, I'm sure, virtually untapped. Education and learning on all levels are wildly in need of a transfusion. The problems are: Can we think and design creatively enough? And can we find a way to pay for it?

REFERENCES CITED AND RELATED READINGS

Barr, Avron, Marian Beard, and Richard C. Atkinson, 1978. "The Computer as a Tutorial Laboratory: The Stanford BIP Project," International Journal of Man-Machine Studies, 8:567-596.

Boden, Margaret A., 1977. Artificial Intelligence and Natural Man. Basic Books, Inc., New York, NY.

Bregar, William S., and Arthur M. Farley, 1980. "Artificial Intelligence Approaches to Computer-Based Instruction," Journal of Computer-Based Instruction, 6(4):106-114.

Brown, John S., Richard Rubenstein, and Richard R. Burton, 1976. "Reactive learning environment for computer-based electronics instruction," Bolt Beranek & Newman Report No. 3314.

Brown, John S., and Richard R. Burton, 1978. "Diagnostic models for procedural bugs based in basic mathematical skills," Cognitive Science, 2:155-192.

Brown, John S., and Richard R. Burton, 1977. "A Paradigmatic Example of an Artificially Intelligent Instructional System," Bolt Beranek & Newman Report No. 152 277.

Burton, Richard R., 1979. "An Investigation of Computer Coaching for Informal Learning Activities," International Journal of Man-Machine Studies, 11:5-24.

Carbonell, J.R., 1970. "AI in CAI: An Artificial Intelligence Approach to Computer-Assisted Instruction," IEEE Transactions on Man-Machine Systems, MMS-11, 4:190-202.

Clancy, William J., 1979. "Tutoring Rules for Guiding a Case Method Dialogue," International Journal of Man-Machine Studies, 11:25-49.

Clancy, William J., James S. Bennett, and Paul R. Cohen, 1979. "Applications-Oriented AI Research: Education." In Barr, Avron, and Edward A. Feigenbaum, eds., Handbook of Artificial Intelligence.

Crowder, Norman A., 1960. "Automatic Teaching by Intrinsic Programming," in A.A. Lumsdaine and Robert Glaser (eds.), Teaching Machines and Programmed Learning: A Source Book, pp. 286-298. Washington, D.C.: National Education Association of the United States.

Gable, Alice, and Carl V. Page, 1980. "The Use of AI Techniques in Computer-Assisted Instruction: An Overview," International Journal of Man-Machine Studies, 12:259-282.

Goldstein, Ira, 1978. "Developing a Computational Representation for Problem-Solving Skills," AI Memo 495, Massachusetts Institute of Technology, Cambridge, Mass.

Heines, Jesse M., 1983. "Basic Concepts in Knowledge-Based Systems," Machine-Mediated Learning, 1(1):65-95.

Pressey, Sidney L., 1926. "A Simple Apparatus Which Gives Tests -- and Teaches," School and Society, 23(586): 373-376.

Pressey, Sidney L., 1927. "A Machine for Automatic Teaching of Drill Material," School and Society, 25(645):549-552.

Sleeman, D. and J.S. Brown, ed., 1982. Intelligent Tutoring Systems. New York: Academic Press.

Smithsonian Institution, 1971. "Technological Augmentation of Human Cognition: An Interdisciplinary Review," Smithsonian Institution, Washington, D.C.

Wallach, Michael A., 1966. "Creativity and the Expression of Possibilities," U.S. Dept. of Health, Education and Welfare, Washington, D.C.

Weizenbaum, Joseph, 1976. Computer Power and Human Reason - From Judgement to Calculation. W.H. Freeman and Company, San Francisco, California.