F O U R   F A C E S   O F   H A L :

A Framework for Using Artificial Intelligence Techniques in Computer-Assisted Instruction

COINS Technical Report 73C-4 (rev.)

By

Howard A. Peelle

and

Edward M. Riseman

University of Massachusetts
Amherst, Massachusetts

January 1974

ABSTRACT

This paper describes a conceptual framework for applying artificial intelligence (AI) techniques in computer-assisted instruction (CAI). The student progresses through four phases ("faces") of interaction with the computer acting as "Benevolent Mentor", "Cognizant tool", "Discerning Partner", and "Learner".

The first phase serves to introduce the problem space, with the computer initiating as well as responding to inquiries. In the second phase, the computer provides the student with a set of tools which magnify his problem-solving ability. The third phase employs the computer as a partner, supporting a joint problem-solving endeavor. Finally, the computer plays the role of a learner, and the student learns by teaching it how to solve new problems.

Each phase has an analog in general problem-solving techniques; namely, problem representation, heuristic search, directed search, and formal algorithmic expression.

Taken together, the four phases comprise an integrated educational system using AI in CAI. The student progressively consolidates his knowledge from previous phases and applies that understanding in new or more advanced ways. The system is illustrated within the problem of space of cryptography, and applications to several other subject areas are cited.

INTRODUCTION

Over the past few centuries, society has been witness to the unsubtle invasion of machines into nearly every aspect of human existence. Computers are now ubiquitous in crucial areas such as transportation, communication, national defense, food and materials production, and scientific research. Today, the development of _intelligent machines_ purports to dramatically affect one of man's most cherished activities: education.

As the twenty-first century rapidly approaches, we advocate _intelligent use_ of intelligent machines, especially in education. Toward that end, this paper explores uses of artificial intelligence (AI) in computer-assisted instruction (CAI). First, we offer a review of both fields of CAI and AI, plus a brief analysis of the evolutionary forces which bring them together.

BACKGROUND: EVOLUTION OF AI IN CAI

The current state-of-the-art of CAI is basically one of prolonged infancy. After an initial upsurge during 1965-1968, research and development activity in the field has subsided. Major impediments to progress have been identified: prohibitive costs and unwieldy technology; low quantity and dubious quality of curricular materials; lack of adequate teacher-training programs; absence of cooperation between profession factions; plus a pervasion of irrational myths and fears (1).

Although CAI has been utilized experimentally at many levels of education in a pot pourri of subject areas, developments over the last five years have not differed significantly from earlier work, and--contrary to predictions--CAI is not widespread in 1974.

In the young field of artificial intelligence, researchers have demonstrated that machines can be programmed to carry out a wide variety of "intelligent" activities--some with potential application in the classroom. For example, programs have been constructed to solve certain algebra word problems (2), perform symbolic integration in the calculus (3), prove theorems in propositional logic (4), complete geometric analogies (5), conduct plausible non-directive dialogue (6), answer questions from an information network (7), and understand natural language statements about an environment of blocks (8).

Until recently, however, research in AI and developments in CAI have gone their separate ways. AI research has been somewhat myopic, conducted with nary an eye out for practical application to pressing real-world problems.

CAI developments have been disappointingly unimaginative, sometimes even stifling and oppressive; in other words, a disproportionately large percentage of CAI applications to date have been lockstep tutorial-drill-and-test sequences in which the student is intellectually and psychologically dominated during instruction.

So, it seems that CAI and AI have evolved to the point where their marriage is imminent. CAI offers AI a challenging arena for testing the effectiveness of sophisticated information processing techniques. AI offers qualities heretofore weak or altogether lacking in CAI--'sensitive' and 'intelligent' interaction with a machine.

FOUR FACES OF HAL: AN EDUCATIONAL SYSTEM USING AI IN CAI

In the minds of many, the IQ of today's typical computer-assisted instruction system is relatively low. After all, there are many questions a young child understands which a computer program cannot sensibly answer. CAI is certainly seen as primitive when compared with HAL, the super-computer portrayed in Stanley Kubrick's futuristic film "2001: A Space Odyssey". HAL not only monitors all flight controls and life support systems on man's voyage to Jupiter but also converses in natural English, competes against the crewmen in games of chess, and psychoanalyzes them to boot!

While such "artificial intelligence" machines are still within the realm of science fiction, application of certain AI techniques can enable a student to engage the computer less as a task-master and more as an interactive resource in an intellectual pursuit. Specifically, we envision an idealized educational system which utilizes the computer in the following roles:

The Computer as Benevolent Mentor

The Computer as Cognizant Tool

The Computer as Discerning Partner

The Computer as Learner

The student progresses through these phases--the "Four Faces of Hal"--consolidating his understandings from each previous phase before applying that knowledge in new, more advanced ways.

To illustrate the dynamics and the pedagogy of this system, we will trace the activities of a hypothetical student in the problem-space of cryptography.[1] Solving a cryptogram is (for most people) a complex task--requiring one to call upon his powers of logic, knowledge of language structures, probability of events, etc. Typically, the problem-solver forms hypotheses, attempts to prove some false by contradiction, and finally develops various heuristics to explore the very large number of possible solutions. There are no known optimal procedures for solving cryptograms.

> Consider the following cryptogram in which each unique letter of the cryptogram represents a unique letter of the alphabet. (The reader is invited to decode the message.)
>
> X Q W V B R Q Q X , D W I D Q B Q M K C D N Z B !
>
> One basic strategy for solving a cryptogram is to attempt to find vowels. As a guide, letters occuring most frequently in the cryptogram can be compared with actual letter occurrences in English, and the most likely substitutions tried first. In the cryptogram above, this strategy suggests assuming Q is E. Another (complementary) strategy explores the short words, looking for possible matches with the collection of known words, such as "a", "I", "the", "to", "of", "for", "and", etc. If the solver is successful at pinpointing some letters in a given word, he can then examine the consequences of those assignments in the rest of the cryptogram. Any contradictions which arise, such as adjacent letters which never occur together in English, reflect back on previous assignments. In this way, hypotheses may be disproved until the solution is reached.

---

[1] Crypotograhy involves enciphering and deciphering messages ("cryptograms") written in secret code. For reference, see Secret and Urgent (9). A cryptographer is one who attempts to discover the hidden message by making certain substitutions for the letters or symbols given. In their recent work, Newell and Simon (10) have studied patterns of problem-solving in the similar area of cryptarithmetic.

For the computer to interact with the student in this problem-space, it must have a basic knowledge of the 'world of cryptograms'. It must have a (partial) dictionary, a table of probabilities of occurrences of letters in English, probability functions, rules of inference, the 'notion' of contradiction, search procedures, and a variety of heuristic strategies. Now let us consider the four phases of HAL in the order that the student meets them.

THE COMPUTER AS BENEVOLENT MENTOR

In the role of "benevolent mentor", the computer acts as a sensitive and flexible tutor. This first phase of interaction provides the beginning student with an opportunity to become acquainted with the problem space at hand. Here, the computer will describe the problem space and typical problem-solving procedures--all at the pace desired and at the level of specificity requested. While the student depends on the computer to present the content, he may influence other factors of instruction, such as sequencing, mode of interaction and level of difficulty.

The computer-as-benevolent mentor assumes that the student may be new to the subject, perhaps even ignorant of basic definitions. The computer invites the student to ask questions and, benevolently, responds with helpful information. In seeking to learn how a cryptogram is solved, for example, the student could ask about the importance of finding vowels, or words ending in "s", or even the occurrence of the article "the".

Alternatively, the computer might ask leading questions or possibly display a sample cryptogram and steps for solving it from start to finish. In any case, of course, the computer is fully apprised beforehand; that is, it must "know" the cryptogram solution, what information is relevant, which strategies are dependable, etc.

Generally, for the computer to be an effective pedagog in this role, it must be capable of: (1) formulating questions and processing responses, (2) receiving questions and producing answers, and (3) changing modes according to the response patterns of the particular human tutee.

Artificial intelligence provides this phase of the educational system with techniques to make this information processing somewhat flexible. Semantic information networks have been used to allow the computer to answer factual questions as well as generate questions and corresponding answers (11) - (13). Carbonell (13) demonstrated the feasibility of such interaction--"mixed-initiative dialog"--between the computer and a student of geography. Similarly for cryptography, we can store in a semantic network the definitions of concepts and characteristics of popular strategies, as well as facts about the English language that we wish to introduce to the student.

This 'face of HAL' differs from conventional CAI in two important respects: (1) the curriculum developer is not required to enter all anticipated student responses nor prespecify all allowable questions; and (2) the student is not expected to remain passively assimilating facts during a lesson, but may <u>initiate</u> inquiry. AI programs of this kind can exhibit some characteristics of a good teacher. If a student does not understand a question or an answer, he can request further explanation; or, if the student is proceeding successfully in the tutorial, the "benevolent mentor" can proceed at a suitably rapid pace. Thus, interaction with a machine programmed in such a way is a flexible, sensitive process.

In terms of general AI problem-solving technique (14), this phase of "HAL" is analogous to problem representation, in which both the states and the operators of a problem space are described. In cryptography, one can think of each subset of letter substitutions as a possible state and the operators as mechanisms by which one state is transformed into another. The "Benevolent Mentor" introduces the student to only one of several choices of problem representation usually available. The one introduced in this first phase forms the vocabulary for the following stages. Education in any problem-oriented domain begins with some problem representation (either explicit or implicit). Since here we are dealing with definitions explicitly and somewhat formally, the result should be less ambiguity later in the communication between student and machine. Ideally, the student is provided very early with a well-organized framework--or language--for thinking about problem solving.

When the student is ready to move on to the next phase, he should have acquired some knowledge of the problem space. He should be familiar with the information which characteristically describes cryptography; he should have identified stages along the way toward solutions of sample cryptograms; and he should have developed some intuitive strategies of his own.

THE COMPUTER AS COGNIZANT TOOL

In the role of "cognizant tool", the computer provides the student with a set of tools which magnify his problem-solving powers. By applying a selected set of tools, the student can explore more deeply and/or more broadly the complex alternatives at any stage in the search for a solution. And, by selecting different tools, he can receive feedback on the utility of various possible strategies for solving the problem at hand.

Such tools are prestored programs which are not only themselves useful for problem-solving but also oversee their own usage. Used in a laboratory-like setting, these tools do not impart knowledge in a pedantic way, but rather they teach in the context of their usage. A tool is "cognizant" to the extent that it can inform its user of how it is being used. If a tool is misused, one is informed of the proper (or the practical) range of usage; if used properly, one is assured (by mute confirmation) that one is not proceeding on the basis of aberrant results.

In cryptography, there are several tools appropriate for problem-solving. One of the simplest tools for cryptogram-solving is a "scratch pad". Here, the computer performs several routine housekeeping functions: it remembers the original coded message; it records the set of assumptions made; and it displays the partially decoded message. Information is updated during problem-solving, hence relieving the user of the need for pencil and paper.

An extremely useful tool in the problem-space of cryptography is "assume". Whenever the "assume" tool is used, the computer carries out the assumption designated; e.g. ASSUME Q IS E would cause the computer to replace all occurrences of Q in the cryptogram with E's.

```
X Q W   V   B R Q Q X,   D W I   D Q   B Q M K C   D N Z B!
_ E _   _   _ _ E E _,   _ _ _   _ E   _ E _ _ _   _ _ _ _!
```

The tool is cognizant of sequences of assumptions and can announce the levels of all currently existent assumptions. Also, during the use of "assume", the tool itself can inform the user if he has--perhaps unknowingly--made an assumption which competes with an earlier one. A complementary tool allows the user to "unassume"; that is, he may negate any prior assumptions or even all assumptions.

Another valuable tool is "imply". This tool yields the implications of any set of assumptions based on information about the English language. If the assumptions regarding a cryptogram reduce the possible identity of other still-coded letters to a smaller subset of the alphabet, the user is so informed. For example, a table of the digrams (letter pairs allowable in English) can be used to reduce the possible choices for coded letters adjacent to already assumed letters. The "imply" tool is particularly valuable if the implications of a given set of assumptions produce a <u>contradiction</u>. When a hypothesis about the cryptogram is so disproved, this tool informs the user, enabling him to eliminate that set of assumptions from further consideration.

Other, more sophisticated tools could utilize statistical data. For instance, following an assumption, a "probability" tool could be applied to help predict likely identities of coded letters from the set of all implications. It could also be used to estimate the likelihood that a given set of assumptions leads to a solution. This tool might draw upon tables containing probabilities of letter combinations occuring in the English language; e.g. letter pairs, letter triplets, prefixes, suffixes, etc. Since the computer does not 'know' the solution to the cryptogram, the user is provided with only an estimate--whose accuracy depends on the degree to which the cryptogram statistics conform to English language standards.

One singularly powerful tool is "search". This tool searches through all possible substitutions for a given subset of coded letters and returns to the user a list of the substitutions which are most likely to succeed. Of course, a search through all possible substitutions for all coded letters, in theory, assures a solution to the cryptogram (if one could recognize it!). But it is misleading to consider this exhaustive search as feasible, since the number of possible solutions may be as large as 26! (approx. 4×10^{26}). The "search" tool, therefore, could use the "probability" tool, based on standard English language statistics. While this tool is being used, it would report to the user how well it is faring. If the subset of coded letters is large, "search" might tell the user approximately how long it will take to return an answer. If after a reasonable computation time no solution is reached, the tool might offer suggestions on how to cut down the search. Or, if the choice of coded letters to be searched is unproductive, the tool might suggest a different subset. In any event, the student-user directs the use of the tool--he has final authority and ultimate responsibility.

Some other conceivable tools include "spot" and "match". The "spot" tool examines the cryptogram and attempts to find patterns--repeating sequences of crypto-letters or words. The "match" tool seeks to find in a list of common words those words which have letters in the same positions as a partially completed word in the cryptogram.[1]

Experience with these tools will allow the student to judge the utility of various tools in subsequent journeys through the problem space. And, by combining the tools in different problem-solving circumstances, he can begin to develop heuristic strategies.

This interaction differs from most convential CAI in two basic ways: (1) greater control for the student, and (2) emphasis on studying procedures. Here the student is controlling the search for the solution. The computer does not know the answer, nor is the computer monitoring the student's work. Clearly the student is responsible for paths chosen.... including blind alleys! Secondly, since there are usually many possible ways to solve a problem, there is no single, "correct" educational objective. Instead, the student actively learns procedures and related heuristics for problem-solving.

In analogy with AI problem-solving technique, this phase of interaction introduces the concept of "heuristic search". Within a problem domain, goals are often sought by applying sequences of operators to a starting state,

---

[1] Preliminary investigation of this tool indicates that it may be sufficiently powerful to solve most cryptograms; that is, after one or two letters are fixed in a long word, like _ _ _ _A_E, the only "match" in a list of about 800 common words is VILLAGE, which immediately gives four additional letters for substitution elsewhere in the cryptogram!

producing new states to be examined. However, in most problems, the number of possible combinations is huge. Thus, the theory of bringing to bear any knowledge--even intuitive shortcuts--in order to reduce the amount of search is apropos. Note, however, that there is no systematic, over-arching organization for conducting the search, except what the student develops.

THE COMPUTER AS DISCERNING PARTNER

In the role of a "partner", the computer provides for man-machine collaboration; that is, the student and the computer jointly engage in problem-solving. Presumably, by this point, both the student and the computer have the necessary vocabulary and knowledge of problem-solving procedures as well as some notions of the effectiveness of various heuristics. Therefore, they can carry on a meaningful (albeit limited) dialogue about the problem at hand.

Man-machine collaboration is most viable in a goal-oriented environment, such as game-playing or problem-solving. Here artificial intelligence techniques can be applied to assist in planning, organizing, and executing complex strategies. Methods of problem-reduction for example, can be employed by computer to aid in finding goals or in designating the most plausible, the most beneficial or the least costly sub-goals.

In the environment of cryptography, the computer-as-partner permits a "discussion" of alternative problem-solving strategies as they are being formulated. This discussion involves both the student and the computer proposing plans of attack, analyzing and commenting on them. Specifically, either might:

- suggest initial goals
- reduce a problem to sub-goals
- formulate integrated strategies
- evaluate the merits of different strategies
- test for solutions
- alternate the initiative

From the student's perspective this means that he can suggest a tentative plan--perhaps representing a sub-goal--and expect to have that plan reviewed and critiqued by the computer (to the extent that it is capable). For example, the student might recommend the sub-goal of determining if one of the two most frequent crypto-letters is E. Based on evidence at the computer's disposal, such as probabilities of all letters in English and tables of allowable letter pairs (or n-tuples) it would respond with an assessment of the likelihood of this plan to succeed.[1]

In addition to reviewing the student's plan, the computer can offer suggestions for reasonable next steps to be taken in solving the problem. If the student pursues another plan, of course, the computer can agree or

---

[1] E is the letter which occurs most frequently in English. If no immediate implications of the particular assumption generate contradictions (such as QE), then it is a reasonable first step to assume that E is one of the more frequently occurring letters in the cryptogram.

disagree, as before. But if the student has run out of conjectures, his discerning computer-partner is prepared to take the initiative. Continuing with the previous example, the computer could then recommend its choice for which of the two most frequent crypto-letters to try assuming first. (The probability tool would be employed by the computer to make this judgment.) Alternatively, if neither letter appears to be E (as determined by application of the implication tool), the student might request suggestions on how to proceed. The computer-partner could then suggest a coherent strategy, such as the one beginning by analyzing short words in order to find other good candidates for the most frequent crypto-letters.

Of course, there exist higher level strategies of which the computer has no working knowledge nor upon which it can comment. The wisdom of pursuing such strategies rests more reliably with the human problem-solver. Typically, the contributions of the human would be at a higher level, say based on complex contextual clues or raw inuition. Generally the partnership seeks to utilize the talents of each system optimally relegating those tasks for which the computer-partner is best suited to the computer and those tasks in which humans excel to the human.[1]

This kind of interaction differs from conventional CAI in its emphasis on developing powers of reasoning and in its promoting "discussion" with an "intelligent" machine. As the student becomes conscious of the bounds and complexities of a problem, and in choosing certain paths toward the

---

[1] (Of course, we are also using the computer-partner in a role for which it is not naturally suited; namely, discussion.)

solution of a problem, he must occasionally defend his decisions to his computer-partner. This promotes a "discussion", limited only by the understandings of each partner in dialog. Ideally, with such a partnership, tools are used more effectively, strategies are formulated more formally and efficiently, and the likelihood of reaching a solution is increased beyond what either could account for alone.

In the AI problem-solving analog, this phase of interaction involves a context-sensitive ordering of operators (tools) in the search through a state-space for a solution--in other words, directed search. AI researchers have developed techniques to reduce problem complexity, such as establishing STRIPS-like (15) "pre-conditions" in order to apply operators only at certain times during a heuristic search. Whereas, in the "cognizant tool" phase, the search control was conspicuously absent, the "computer-as-partner" facilitates discussion of the relative merits of setting up subgoals in particular contexts that arise during the problem-solving activity.

THE COMPUTER AS LEARNER

In the role of "learner", the computer must be capable of receiving instruction; that is, it must be teachable.[1] In the problem space of cryptography, the computer-as-learner provides the student with a meta-language for teaching -- one in which he may express formal procedures for solving cryptograms. Treating the computer as initially "unknowledgable", the student's goal is to teach the computer how to solve cryptograms. Of course the computer is not totally ignorant to begin with; the meta-language embodies the basic vocabulary of the problem space and the complete set of problem-solving tools used in the previous phases. But the computer knows nothing about how to put this information together to solve cryptograms; that is, it is ignorant of strategies.

It is the student's task to construct a self-contained procedure which will solve cryptograms not previously encountered. This procedure may contain alternative strategies and means for analyzing problem goals and subgoals; it may utilize any of the available tools and any heuristics the student can express explicity. For example, the student might begin by articulating discrete steps to be followed, like: "First, assume Q is E; then, if there are no contradictions, try assuming THE for the first three-letter crypto-word ending in E, etc." These steps are translated by the language processor of our computer-as-learner -- operating

---

[1] By 'teaching a computer' we mean more than the routine use of a programming language to execute computational commands. To be sure, when one writes a program, he is teaching the computer what to do, and in this respect the computer acts like an ideal tutee: it requires explicit expression and acts only on information provided. But, we propose that the programming language be supplanted by a higher-order language -- a "meta-language"-- which enables the student to express strategies for problem-solving.

in a well-circumscribed micro-world -- into a self-contained, excutable algorithm which is then applied to a cryptogram.

Each application of the algorithm to a new cryptogram tests how well the computer has been taught. If the algorithm is successful, the student receives assurance that his teaching approach was effective. If, however, the algorithm proves no longer successful, the student may discover that the strategy he taught the computer is not flexible or not general enough. At this point, it is incumbent upon the student to retrain the computer. Modifications or futher instruction may be warranted, like changing the overly specific first step (above) to: "First assume that the most frequent crypto-letter is one of the six most common letters in English (E T A O I N) and check each for implications; proceed with the substitution having the highest probability of success; if all subsequent attempts at the solution fail, back up and try the substitution with the next highest probability."[1] Eventually, after successive phases of refinement, the computer's algorithm for solving cryptograms becomes expressed in more generalized terms and works for more cryptograms presented.

This approach differs from conventional CAI in a fundamental way: it places the student in a teaching position! Not only does the student learn _by_ teaching (N.B. many teachers claim that they did not "really learn" their subject area until they had to teach it) but also he learns _about_ teaching, a complex process in itself. More specifically, this learning on the part of the student occurs during each of two activities: (1) in _describing_ procedures, and (2) _in observing_ the consequences of his teachings. The first involves learning to express problem-solving

---

[1] A computer programming language such as PLANNER [16] is well-suited for this kind of automatic back-tracking.

procedures formally -- i.e., being rigourous and logically consistent in their description -- and to confront directly the concept of generalization. Secondly, the student gains an increased awareness of educational processes, for in teaching the computer, he must review his own knowledge, decide what is to be taught and how it is to be taught.

From an AI perspective, this fourth phase parallels problem-solving technique in its strife for formal expression of automated search algorithms. This is analogous to the problem of designing a self-contained problem-solving machine (within the constraints of the meta-language provided) for searching through an incompletely understood state-space with the intent of increasing the likelihood of success in a generalized algorithm.

In sum, the computer-as-learner phase provides a rich and stimulating environment which brings teaching and learning into sharp focus. Our belief is centered on the notion that analysis, synthesis, and insight into knowledge all take place when one attempts to teach another.

SUMMARY: GENERALITY AND EXTENSIONS TO OTHER DISCIPLINES

We have described a design for an educational system -- the "FOUR FACES OF HAL" -- which can be viewed as a general framework for utilizing AI in CAI. Conceptually, this system is analogous to problem-solving techniques. The first "face" is equivalent to the pedagogical first stage of learning about a particular problem area -- or, representing the problem space. The second face uses tools for learning -- or, the operators for manipulating states in a heuristic search. The third face engages the computer as a partner in problem-solving -- or, jointly directing a context-sensitive search of the state-space for a solution. The fourth face transcends the previous set of three faces in equipping the computer with learning capabilities -- or, translating problem-solving procedures into formal algorithms.

Educationally, this system is qualitatively different from conventional CAI and may conceivably extend to other disciplines. In high school geometry or trigonometry, for example, one might construct a set of tools which allow the student to direct the computer to manipulate a diagram and, perhaps, ultimately learn proofs of properties of geometric figures. Or, in calculus a student attempting to integrate a difficult function might be able to hold a productive discussion with his computer-partner concerning, say, the plausibility of integrating by parts as opposed to using certain substitution (cf. Slagle's SAINT [3]). Similar potential for using AI in CAI appears in problem spaces other than mathematics, such as physics, medicine, and even language arts.

CONCLUSION: SOBERING PRACTICALITIES

In closing, we offer some sobering comments on practicalities. This paper has dealt primarily on what can be rather than what is. It is easy to paint educational utopias, but we do not wish to play ostrich to problems impeding the development of the ideas we put forth. Some of the difficulties we feel compelled to identify are present in the form of technical problems, economic constraints, and unresolved issues.

First, technical problems loom and lurk about. All of the approaches described herein fall within the state of the art in both AI and CAI. While these approaches can be implemented now, we do not wish to leave the reader with the impression that the necessary developments are trivial. For instance, the technical problems inherent in approaching natural processing are formidable indeed.[1] Ideally, unrestricted conversation is desirable in a CAI system, but we are a long way from that!

Secondly, economics constrain the use of AI in CAI today. CAI developments have proved prohibitively expensive in the past, and, even though technological advances have begun to lower hardware prices, costs of curriculum development, distribution, and training remain high. Most research endeavors in AI also require sophisticated software and equipment, with correspondingly high-grade prices.

---

[1] There are difficulties in applying Winograd's (8) approach to domains of discourse other than that of the "blocks world"; the vocabulary, procedures, and goals -- in short, the computer's complete knowledge of its environment -- would have to be respecified. Additionally, the program requires a moderately large amount of storage space devoted to the language processor, leaving little room for "curriculum"; and, responses are not as fast as students might like.

Other sobering problems emerge as unresolved issues. One issue conjures up the Orwelliam spectre of a sterile, depersonalized society where people are controlled by machines through CAI. The use of AI in CAI does carry the potential danger of behavioral control. Of course, all technologies invented by man are double-edged; that is, they can be used to man's betterment or to his destruction!

Another related issue pitches personal privacy versus data banks. The portent of complete educational monitoring arises when all student records are on central file -- every test score, every evaluation, every academic qualification (or lack of qualification) associated with an individual could be used to automatically and 'objectively' funnel his educational career. This is tracking in its most nefarious form! Society as a whole is now just beginning to grapple with complex questions about data banks, and there do not seem to be any panaceas here.

Long-term philosophical issues also become heated when the efficiency of educating via computer systems violates humanistic concerns. In some views, CAI evokes the image of the perfect teacher: relentlessly patient, exceedingly accurate, honest and fair, systematically ensuring learning for all students. For others, the machine is a vulgar instrusion into a domain exclusively research for human sensitivity: teaching. "No computer can match the warmth and caring of a human", they would argue. Whether these questions are answerable or not, they should be discussed and the implications of "intelligent" CAI examined.

(16) C.E. Hewitt, "PLANNER", MAC-M-386, MIT, Oct. 1968; revised June 1969.