4. An investigation of computer coaching for informal learning activities

RICHARD R. BURTON AND JOHN SEELY BROWN

Bolt Beranek and Newman, Inc., 50 Moulton Street, Cambridge, Mass. 02138, U.S.A.

Computer-based tutoring/coaching systems have the promise of enhancing the educational value of gaming environments by guiding a student's discovery learning. This paper provides an in-depth view of (i) the philosophy behind such systems, (ii) the kinds of diagnostic modeling strategies required to infer a student's shortcomings from observing his behavior and (iii) the range of explicit tutorial strategies needed for directing the Tutor to say the right thing at the right time. Examples of these issues are drawn for a computer-based coaching system for a simple game–How the West was Won. Our intention in writing this paper is to make explicit the vast amounts of tutorial knowledge required to construct a coaching system that is robust, friendly and intelligent enough to survive in home or classroom use. During the past three years, we have witnessed how subtle the computer-based coaching problem really is. We hope this paper conveys some of these subtleties—many of which continue to resist general solution.

Introduction

The revolution in personal computing will bring with it extensive use of complex games. Students will play computer-based games during much of their free time. These activities can provide rich, *informal* environments for learning. Games provide an enticing problem-solving environment that a student explores at will, free to create his own ideas of underlying structure and to invent his own strategies for utilizing his understanding of this structure. Properly constructed games can lead to the formation of strategies and knowledge structures that have general usefulness in other domains as well. However, a major stumbling block to the effective educational use of unstructured gaming or open-ended problem-solving environments is the amount of tutorial resources that are often required (i) to keep the student from forming grossly incorrect models of the underlying structure of the game/environment, (ii) to help him see the limits of his strategies, and (iii) to help him discover the causes of manifested errors.

One of the prerequisites for a productive informal learning environment is that it be made enticing to the student by enabling him to control it. The student must have the freedom to make decisions (incorrect as well as correct ones) and observe their results. While a student's incorrect decisions sometimes lead to erroneous results that he can immediately detect, they often produce symptoms that are beyond his ability to recognize. For an informal environment to be fully effective as a learning activity, it often must be augmented by tutorial guidance that recognizes and explains weaknesses in the student's decisions or suggests ideas when the student appears to have none. This is a significant challenge requiring many of the skills analogous to those of a coach or

† This research was supported, in part, by the Advanced Research Projects Agency, Air Force Human Resources Laboratory, Army Research Institute for Behavioral and Social Sciences, and Navy Personnel Research and Development Center under Contract No. MDA903-76-C-0108.

laboratory instructor. The tutor or coach† must be perceptive enough to make relevant comments but not so intrusive as to destroy the fun inherent in the game. This paper presents one such coaching system (named WEST) built around the game "How the West was Won." The system is examined as an instance of a general paradigm, called "Issues and Examples," for building such systems. Aspects of the system are examined to discover the limitations of the central paradigm and to characterize a wide variety of tutorial strategies that must be included to create a successful coaching system.

COACHING INSTUCTIONAL SYSTEMS

The pedagogical motivation underlying much of our coaching research can be characterized as "guided discovery learning." It assumes a *constructivist* position, in which the student constructs his new knowledge from his existing knowledge. In this theory, the notion of misconception or "bug" plays a central role. Ideally, a student's bug will cause an erroneous result that he will notice. If the student has enough information to determine what caused the error and can correct it, then the bug is referred to as *constructive*. If, however, the student does not have sufficient information to change his behavior as a result of the perceived error, the bug is termed *non-constructive*. One of the most important aspects of a learning environment is the degree to which the mistakes that a student makes are constructive. (See Fischer, Brown & Burton, 1978 for further discussion.) From this point of view, one of the major tasks of a Coach is to give the student additional information in order to transform non-constructive bugs into constructive ones. An additional task for the Coach, in dealing with bugs that do not have easily observable manifestations, is to point out that something can be improved.‡

A subtle requirement of this theory is that the Coach does not interfere too much. While the student is making mistakes in the environment he is also experiencing the idea of learning from his mistakes and discovering the means to recover from his mistakes. If the Coach immediately points out the student's errors, there is a real danger that the student will never develop the necessary skills for examining his own behavior and looking for the causes of his own mistakes.

There are two major but related problems that must be solved by a computer Coach. They are:

(1) when to interrupt the student's problem solving activity, and
(2) what to say once it has been interrupted.

In general, solutions to these problems require techniques for determining what the student knows (procedures for constructing a diagnostic model) as well as explicit tutoring principles about interrupting and advising. These, in turn, require a theory of how a student forms abstractions, how he learns, and when he is apt to be most receptive to advice. Unfortunately, few, if any, existing psychological theories are precise enough to suggest anything more than caution. The requirements that evolve from designing coaching systems should provide useful goals or forcing functions for future cognitive

† This usage of the term "coach" was originated by Goldstein (1977). We originally conceived of the West tutorial resource as a congenial "tutor" but the images evoked by the term "tutor" have proven to be inappropriate. In this paper we shall use "coach" to emphasize the informal nature of the learning situation.

‡ In a recent paper on the educational implications of Piaget's psychological theory, Groen has identified similar requirements. "A child will learn only if he extends the range of hypotheses he can generate and modifies or eliminates the transformations that lead to false ones. Thus, it is part of the teacher's task to ensure that the child is aware of anomalies and counter-examples that result from his activities" (Groen, 1978).

theories. In addition, the coaching systems themselves should be good test environments for such theories.

DIAGNOSTIC MODELING

Since the student is primarily engaged in a gaming or problem-solving activity, any explicit diagnosing of a student's strengths and weaknesses must be unobtrusive or subservient to his main activity. This means that the diagnostic component cannot use prestored tests or pose a lot of diagnostic questions to the student. Instead, the computer coach must restrict itself mainly to inferring a student's shortcomings from whatever he does in the context of playing the game or solving the problem. This can be a difficult problem. Just because a student does not use a certain skill while playing a game does not mean that he does not know that skill. For example, an opponent may never have created a situation that required him to invoke it. Although this point seems quite obvious, it poses a serious diagnostic problem. The absence of a manifested skill carries diagnostic value if and only if an expert in an equivalent situation would have used that skill. Hence, apart from the outright errors, the main window a computer-based Coach has to a student's misconceptions is through a "differential" modeling technique that compares what the student is doing with what the expert would be doing in his place. (See Sleeman & Hendley's article in this issue for further discussions on this point.) This "difference" must provide hypotheses about what the student does not know or has not yet mastered.

The process of constructing a differential model requires two tasks—both of which use a computer-based Expert,† but for different purposes. The first task is evaluating the quality of the student's current action or "move" in relationship to the set of possible alternative moves that an Expert might have made in the exact same circumstances. The second task is determining the underlying skills that went into the selection and composition of the student's move as well as each of the "better" moves of the Expert. In order to accomplish the first task, the Expert need only use the *result* of its knowledge and reasoning strategies, which is in the form of better moves. However, for the second task, it has to consider the "pieces" of knowledge involved in selecting and generating the better moves, since the absence of one of these pieces of knowledge might explain why the student failed to make a better move.

FORMS OF DOMAIN EXPERTISE FOR COACHING

The representation of domain expertise in a computer can be in one of two forms. One form is as a "glass-box" or articulate model (Goldstein & Papert, 1977). The model is referred to as "articulate" because each problem-solving decision it makes can, in principle, be explained in terms that match (at some level of abstraction) those of a human problem-solver.‡ In contrast to the articulate Expert is the "black-box" Expert, which has data structures and processing algorithms that do not mimic those used by human beings. For example, the circuit simulator underlying SOPHIE-1 (Brown & Burton, 1975) is a black-box Expert, and is used only to check the consistency of student's hypotheses and answer some of his questions. Its mechanisms are never revealed to the student since they are certainly not the mechanisms the student is expected to acquire.

† From here on, the term Expert will be used to refer to the simulation of an expert player in the computer.

‡ The BUGGY (Brown & Burton, 1978), WUMPUS (See Goldstein's article in this issue), and GUIDON (see Clancey's article in this issue) systems are based on articulate experts, as are many production rule based experts.

Within the framework of the diagnostic problems faced by the computer Coach, the glass-box Expert seems to be the most useful since it can be used both for the evaluation process (by generating optimal moves) and for determining the skills underlying those moves. Skill determination is achieved by looking at the Expert's problem-solving trace for generating a given move and noting the skills that it used. The glass-box Expert is also useful in the evaluation task because it can generate the space of alternative "better" moves and hence determine the rank ordering of the given move. Note, however, that since the evaluation process involves determining the complete range of alternative behaviors, it requires substantially more computation and robustness than simply assessing the skills underlying any one particular move.

Since the implementation of a black-box Expert is not constrained by human-like algorithms, it potentially can be considerably more efficient and, therefore, more useful for evaluation of a student's move. However, the skills it uses to generate an optimal move are not analogous to the student's, so it can not be directly used for the skill determination task. This raises the possibility of combining an efficient and robust black-box Expert for evaluation with a less efficient glass-box Expert for skill determination.

Computational efficiency is not the only reason for developing the interplay of these two forms of expertise. The black-box Expert used for evaluation need only be augmented with those *incomplete* pieces of an articulate Expert which are needed to detect critical or tutorable *features* of the answers produced by the black-box Expert. The glass-box Expert need not be able to produce the complete solution itself. It needs only to work backwards from the solution to determine the "important" (tutorial) features of the solution. This realization opens up the possibility of constructing coaching systems for domains for which we do not have complete glass-box expertise.

It is possible that a lot of informal learning occurs through the combination of tacit expertise (in the form of a black-box) with incomplete but articulate pieces of a glass-box Expert. For example, no one has a complete, articulate theory on how to play expert chess. Although there are some excellent chess machines, they rely on non-human strategies for achieving their expertise, that is, they are black-box experts. There are also handbooks of chess principles which reflect pieces of articulate knowledge about opening moves, end game tactics, etc. A chess Coaching system could take advantage of the black-box Expert to identify critical moves and use incomplete but articulate knowledge to partially explain why the move was critical and how it might have been detected. People appear to learn natural language through a similar interaction. A complete, articulate theory of English does not exist. People do, however, manage to become fluent in English by receiving feedback from many "black-box experts"—other people who speak it. To help in the critiquing task, there are incomplete articulate pieces of knowledge, such as subject–verb agreement. That is, in addition to getting black-box feedback of the form "that's not grammatical," which could mean almost anything, people also get glass-box rules such as "Don't say 'they is,' say 'they are,' because you must have subject-verb agreement."†

† In this case, it might seem that the black-box Expert plays no significant role since the pieces of articulate knowledge used to critique the sentence could also be used to perform the role of the black-box; namely, reject the sentence as being ungrammatical. However, the black-box Expert also uses tacit knowledge to analyze the sentence in order to isolate structural elements (e.g. nouns, verbs) which are required for the articulate mini-theories or principles. We all know the subject-verb agreement rule and are very skilled at recognizing nouns in sentences, but very few of us can articulate a precise definition of a noun.

The modeling technique discussed in this paper employs a black-box Expert in conjunction with a set of local glass-box Experts. Briefly, the black-box Expert is used to determine the range of possible moves the student could have made, and the glass-box chunks of expertise determine possible causes for the less than optimal behavior of the student. As such, we hope this technique might also be useful in providing insights into how to transform various black-box Experts that currently exist (such as the symbolic integration capabilities of MACSYMA) into interesting, educational systems.†

TUTORING BY ISSUE AND EXAMPLE—A GENERAL PARADIGM

To be played well, any game complex enough to be interesting requires many different skills. From the point of view of a Coach, this is an important fact because it means that when a student does not perform well in a particular situation, it is not necessarily clear what skill he is lacking. The difficulty of determining which skill is being misused is increased by the fact that much of the evidence that the Coach has is indirect. That is, the Coach only knows that the student did *not* make a better move. From this negative information, he must determine why not, i.e. the move itself does not manifest a symptom or an error but the *absence* of another move does. (Contrast this with the subtlely different situation confronting BUGGY in which a bug in a kid's subtraction procedure will have symptoms explicitly contained in the BUGGY answer.)

OVERVIEW

The paradigm of "Issues and Examples" was developed to focus a coaching system on relevant portions of student behavior and to provide an overall coherence (goal) to the Coach's comments. The important aspects of the domain—that is, the skills and concepts the student is expected to master—are identified as a collection of "Issues". The Issues determine what parts of the student's behavior are monitored by the Coach. Each Issue represents an articulate mini-theory (a piece of a glass-box Expert) concerning the structure of the domain. It is characterized by two procedures. The first watches the student's behavior for evidence that the student does or does not use its particular concept or skill. As such, it is called an Issue Recognizer. The Recognizers are used to construct a "model" of the student's behavior. The second procedure of an Issue knows how to use various parts of the student model to decide if the student is "weak" in that Issue. It is called an Issue Evaluator. Thus each Issue has associated with it both a Recognizer and an Evaluator as procedural specialists.

At any point in the game, the hypotheses concerning the weaknesses of the student can be determined by running all of the Issue Evaluators on the model. When the student makes a "poor" move, his weaknesses are compared with the Issues necessary to make better moves in order to try to account for why he did not make a better move. That is, the Coach looks for an Issue in which the student is lacking and which is required for the Expert's better moves. Once an Issue has been determined, the Coach can present an explanation of that Issue together with a better move that illustrates the Issue. In this way, the student can see the usefulness of the Issue at a time when he will

†The technique might also be useful when there exists a complete glass-box Expert that can not do the problem in "all" ways. For these domains it cannot be assumed that the student is in fact working the same way as the expert.

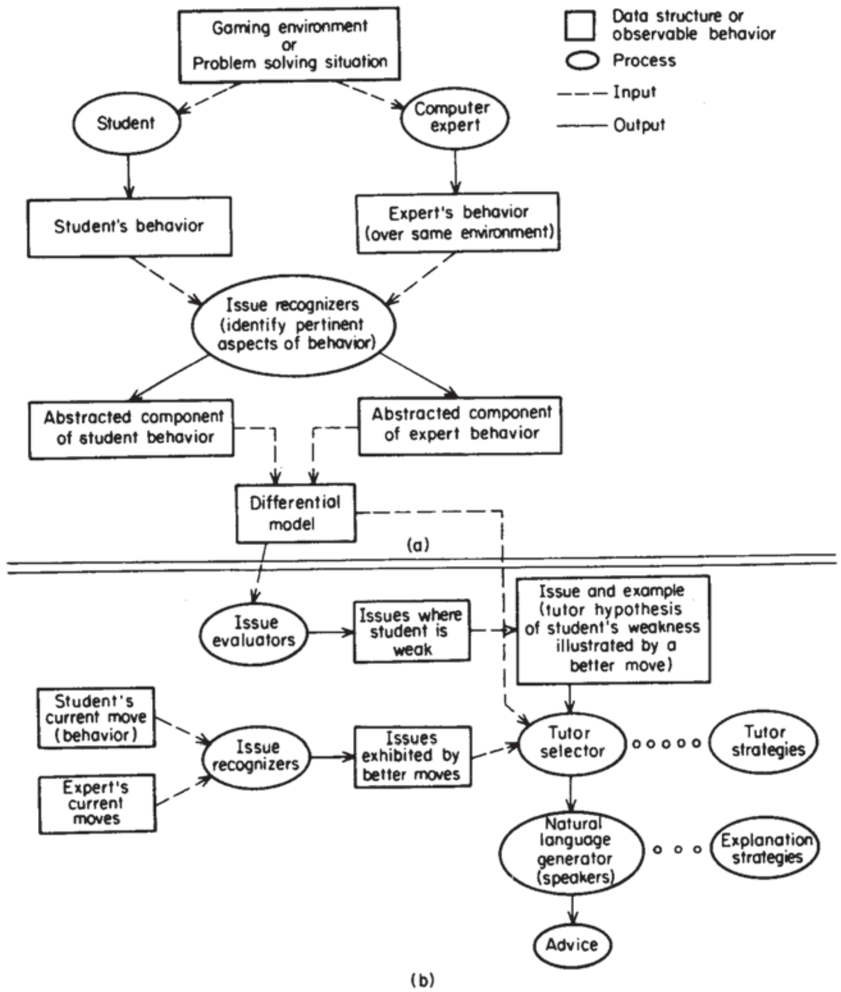

FIG. 1. Information flow diagram of modeller/tutor.

be most receptive to the idea presented—immediately after he has attempted a problem whose solution requires the Issue.

Figure 1 is a diagram of the modeling/tutorial process underlying the Issues and Examples paradigm. Figure 1(a) presents the process of constructing a model of the student's behavior. The model is a summary of the student's performance while solving a series of problems (in this case, moves in a game). Each time the student makes a move, the important aspects of his behavior (the Issues) are abstracted by the Recognizers. This abstracting is also done over the behavior of a computer-based Expert in the *same* environment by the *same* recognizers. The two abstractions are compared to provide a *differential model* of the student's behavior, which indicates those Issues on which the student is weak. We reiterate that without the Expert it is not possible to determine whether the student is weak in some skill, or whether the skill has not been used because the need for it has arisen infrequently in the student's experience.

Figure 1(b) presents the top level of the Coaching process. When the student makes a less than optimal move (as determined by comparing his move with that of the Expert), the Coach uses the Evaluation component of each Issue to create a list of Issues on which the student is weak. With the Expert's list of better moves, the Coach invokes the Issue Recognizers to determine which issues are illustrated by better moves. From these two lists (the "weak" Issues and the "better move" Issues), the Coach selects an Issue and a good move that illustrates it, (i.e. creates an example of it) and decides on the basis of other tutoring principles whether or not to interrupt.† If the Coach decides to interrupt, the selected Issue and Example are then passed to the explanation generators, which produce the feedback to the student.

The gaming situation

"How the West was Won" (WEST) is a computer board game that was originally designed at Project PLATO‡ to give students drill and practice in arithmetic. The board (see Fig. 2) is 70 spaces long. In a turn, each player receives three numbers (from spinners), which must be used in an arithmetic expression (using the operations addition, subtraction, multiplication, and division as well as parentheses) with the constraint that no operator or number can be used more than once. The value of the expression is the number of spaces the student is moved along the board. The object of the game is to be the first player to land exactly on 70. To make the student's task more complicated than just making the biggest number, there are several kinds of special moves. Towns occur every ten spaces. If you land on one, you advance to the next one, There are also shortcuts. If you land on one of these, you advance to the other end of the shortcut.§ And if you land on the space your opponent is occupying, he is bumped back two towns, unless he is on a town. The spinner values in WEST are kept small, so that special moves will often be better (get one further ahead) than making the biggest number.||

Figure 2 shows a board situation that illustrates some of the complexities of tutoring, even in this simple game. The student is at 38, his opponent is at 39,¶ and with his spinners (2, 1, 2), the student makes the expression $2+1\times 2$, resulting in a move of 4. Consider the alternative moves the student could have made: he could have moved 1 and bumped his opponent; he could have moved 2 and landed on a town; he could have moved 6 and taken a shortcut. What possible reasons may underlie this suboptimal move?

† If there are no Issues in common between the two lists, the reason for the student's problem lies outside of the collection of Issues, and the Coach says nothing.

‡ The PLATO game was designed by Bonnie Anderson in Dr Robert Davis's Elementary Mathematic Project (Dugdale & Kibbey, 1977).

§ In Fig. 2, Spaces 5, 25 and 44 are the beginning of shortcuts.

|| The rules assumed in this paper are the ones used on the PLATO system as of 1975. Our Coach system, WEST, allows the student to change many of the rules. For example, the board length, the distance between towns, the location and number of shortcuts, and the set of legal arithmetic operations can all be changed and the Coach will continue to work. In addition, the number of spinners can be changed, but we have not built an Expert for such. Changing the rules gives students the opportunity to see the relationship between the rules and the "feeling" of the game.

¶ WEST is typically used by one student playing against the computer's Expert. It is also possible for two students to play against each other, in which case differential models are constructed for each student, thereby enabling coaching for both players.

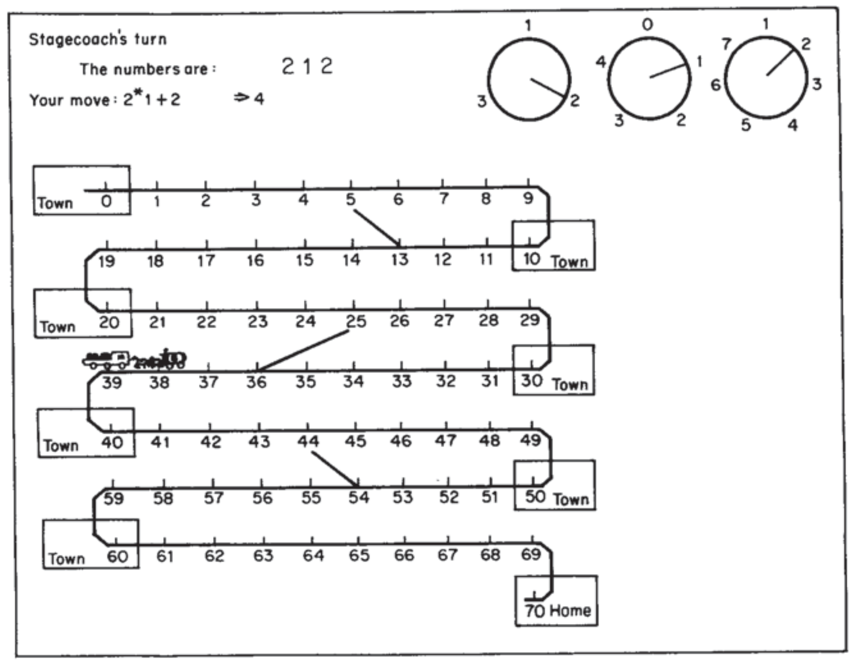

FIG. 2.

THE ISSUES IN WEST

In the Issues and Examples paradigm, the Issues embody the important concepts underlying a student's behavior and define the space of concepts the Coach can address. In WEST, there are three levels of Issues that a coach can focus on. At the lowest level are the basic mathematical skills that the student is practicing. In the current system these include the use of PARENTHESES, the use of various arithmetic operations such as SUBTRACTION and DIVISION, and the form of the student's move as an expression (PATTERN).

The second level concerns the skills needed to play WEST. The Issues at this level are: the special moves of BUMP, TOWN and SHORTCUT; the direction of a move (for example, both FORWARD and BACKWARD are legal); and the development of a STRATEGY for choosing a move, such as maximizing the distnace you are ahead of your opponent.

At the third level are the general skills of game-playing. One such general skill is the strategy of watching your opponent in order to learn from his moves. Another is the effect that different rules of the game have on determining the best strategy.†

† At present the Coaching system does not address these directly.

Each of these Issues is represented in two parts: a Recognizer and an Evaluator. The Recognizers are data-driven from the local context of the student's and the Expert's moves. The Evaluators are goal directed (what are the student's weaknesses?). The Issue Recognizers of WEST are straightforward, but are, nevertheless, more complex than simple pattern matchers. For example, the Recognizer for the PARENTHESIS Issue must determine not only whether or not parentheses are present in the student's move (a lexical check of the expression underlying his move) but also whether they were necessary(which requires parsing the expression) or if they were necessary in the optimal move (which requires parsing the expert's behavior).

For the situation shown in Fig. 2, the following Issues are involved in better moves: Moving 1 entails knowing about the BUMP rule and using SUBTRACTION or DIVISION.† Moving 2 entails DIVISION, knowing about TOWNS, and knowing that the *order* of numbers in the expression does not have to be the same as the spinners. Moving 6 entails PARENTHESES and knowing about SHORTCUTS.

THE MODEL IN WEST

Figure 3 shows some of the fields of a student model created by the differential modeler. The fields it shows include patterns of moves used by the student, special moves, parenthesis usage, and strategy considerations. The columns headed by "MISS" or "MISSED" are places where the Expert would have used the skill but the student did not. They are indications of potential weaknesses. The student shown in Fig. 3 appears to be weak in the Issues PARENTHESIS and BUMP.

TUTORIAL CONSIDERATIONS

Even when relevant Issues and Examples have been identified, it may be inappropriate to tutor. This is determined by invoking various tutoring strategies. One example is the

	(A+B)*C	(A*B)+C	(A+B)-C	(A*B)/C	A/(B*C)	A*(B-C)	(A*B)-C
Moves	1	3	-	-	-	-	-
Best	-	1	-	-	-	-	-
Fair	-	1	-	-	-	-	1
Poor	-	1	-	1	-	-	-
Miss	2	1	1	1	1	1	-

SPECIALS	Took	Missed
Town	3	2
Bump	-	1
Shortcut	-	-

DIRECTION	Good	Poor	Was/best
Forward	5	4	9
Backward	-	-	-

SPIN-ORDER	Good	Poor
Original	2	3
Reverse	-	1
Decreasing	1	-
Increasing	-	-
Other	2	-

	Need & Use	Use Any Way	No Use	None & Miss
(O)'s	1	-	8	4

	Max Delta	Max Dist	Max Number	Specials	Other
Strategy	3	4	2	3	4

FIG. 3. Student model.

† The student could, of course, move 1 without being aware that it will lead to a bump. One ramification of inadvertent moves is that the Model will contain some "noise". Noise will be discussed in the section on Modeling Methodology.

decision about which of the competing Issues to choose. If there are two Issues, both applicable to a certain situation, which should be picked? This is one of the places where a "syllabus" (Goldstein, 1977) might be useful to provide relative orderings of importance or prerequisite links over the space of Issues. However, the Issues in WEST are sufficiently independent that there is little need to consider the prerequisite structure. Instead, additional tutoring principles must be invoked to decide which one of the set of applicable Issues should be used.

We have experimented with two alternative principles for guiding this decision thus far. The first is the *Focus* strategy, which ensures that if everything else is equal, the Issue chosen is that which was most recently discussed; that is, have the Coach hammer away on a particular Issue until it is mastered. The alternative principle is the *Breadth* strategy, which ensures that if everything else is equal, an Issue is selected that has not recently been discussed. This strategy minimizes the chance that a student gets bored by hearing too much about one Issue. A simple agenda mechanism enables either a pure Breadth or Focus strategy.† The default is the Breadth strategy, because it prevents one of two interdependent Issues from blocking the other. Strategies for manipulating the agenda mechanism provide only one source of guidance for the tutor. Additional tutoring principles will be examined in the next main section.

EXPLANATION

Once the decision has been made to tutor on a particular Issue with a particular Example, the Coach still has to decide how to express the Issue to the student. This is the explanation problem. It is in general very difficult. In addition to saying the things the student does not know, conversational postulates dictate that things the student knows already should *not* be said. (See Clancey's article for more of a discussion on this point.) In designing WEST, we have concentrated on the student modeling task and the task of determining when to break in, and have progressed very little on the explanation problem. Currently, the explanations are stored in a procedure attached to each Issue, called a Speaker. Each Speaker is responsible for presenting a few lines of text explaining its Issue. At present, the Speakers work by randomly selecting prestored comments. Several improvements should be made to the Speakers. For example, the explanation should be able to handle multiple Issues. It may be very difficult to distinguish between two Issues, and having a Speaker that can assimilate both into one succinct comment conveniently sidesteps the need to differentiate between them. For example, the Issues of SUBTRACTION and moving BACKWARDS often occur together and it is sometimes difficult to separate the two.

USES OF ISSUES

While the Issues were originally conceived of as guides for the critiquing component of the system, they have proven to have other tutorial uses in our system. One example is when the student asks for help while considering what move to make. If the best move involves an Issue on which the student, is weak, the "hint" can stress that Issue. Our

† The agenda mechanism is implemented as a priority list, along with procedures for reordering it. When two Issues are possible, the one that occurs first on the list is chosen. The "focus" strategy moves a selected Issue to the front of the list, making it more likely to be chosen again, and the "breadth" strategy moves the selected issue further down the list. Since this list can be partitioned into sublists, it is straightforward to have one strategy manipulate the sublists and another to manipulate the elements within a sublist.

motivation here is that the Issue may be the critical piece of information to enable the student to see how he could make the good move, and hence the hint should put emphasis on it.

Issues are also useful in determining when to give the student *positive encouragement*, thus keeping him from viewing the Coach as being only critical. Our current Encouragement strategy directs the Coach to congratulate the student on his good move whenever it is the optimal move that demonstrates an Issue on which the student is weak. However, as we explain next, no one strategy determines what the Coach will do because different strategies may set up competing goals.

Pedagogical strategies

There are many principles that spell the difference between success and disaster in a computer-based gaming-plus-coaching environment for informal learning. Over the last few years, we have had a chance to experiment with WEST and modify it in response to various subtle and not-so-subtle difficulties that we have encountered. In this section we will discuss some of the principles that we found important to embed in our system and identify those which have general applicability to informal learning situations. For the purposes of our discussion, we will distinguish two types of principles—those for structuring the gaming environment itself and those for guiding the Coach within the environment. Although much of what we have discovered concerns explicit learning environments, we believe that many of these principles are also of importance in designing other "friendly" man–machine systems where the feel or ambiance of the total environment (including peripheral assistance or tutoring) is crucial.

PHILOSOPHY

Before discussing these principles, let us briefly summarize the philosophical underpinnings of coaching environments. In these environments it is best for the student to discover for himself as much of the structure of a situation as possible.† Every time the Coach tells the student something, it is robbing him of the opportunity to discover it for himself. Many human tutors interrupt far too often, generally because of a lack of time or patience, and they may be preventing the development in their students of important cognitive skills—the cognitive skills that allow students to detect and use their own errors.

However, there are times when interference with the student's discovery process is called for. In gaming situations, an untutored (unwatched) student may fixate on a subset of the available moves and hence miss the potential richness of the game. In WEST, for example, a student may adopt the strategy of adding the first two spinners and multiplying the result by the third spinner, $(A+B)*C$. Since the third spinner tends to be largest, this strategy is close to the strategy of multiplying the largest number by the sum of the other two numbers (which produces the largest possible result). A student can remain at this plateau indefinitely without perceiving the failings of his strategy. But notice how much of the structure of the game is being missed. The student is unaware of special moves, such as bumps, and therefore of such questions as, "Is it

† This is not to say that structured material (e.g. textbooks) should not have a role in formal education or that guided discovery learning is the only way to learn!

better to send my opponent back 14 or get 9 ahead of him?" Since his strategy does not require searching to determine a move, the student misses the whole notion of strategy as a method for deciding between alternative moves. From the point of view of practicing arithmetic, he is performing one calculation per move instead of the dozens of mental calculations he would have to perform to answer questions such as, "What numbers can I form with these spinners?" or "Can I make a 15 with 9, 10, and 6?" By interjecting comments and suggesting better moves, a Coach can greatly expand the student's involvement in the environment.

The top-level goal driving the Coach is to ensure that its comments are both *relevant* and *memorable*. The Issues and Examples tutoring strategy provides a framework for meeting these two constraints. The Issues are used in the diagnostic process to identify at any particular moment what is relevant. The Examples provide *concrete instances* of these abstract concepts. Providing both the description of the generic Issue (a concept) as well as a concrete example of it increases the chance that the student will integrate this piece of tutorial commentary into his knowledge.

The Issue that is raised must be one in which the student is, in fact, having a problem, lest the advice be ignored or meet with hostility.

> Principle 1: Before giving advice, be sure the Issue used is one in which the student is weak.

The primary ramification of this principle is in how the Evaluators use the student model. As will be discussed in the next section, there is "noise" inherent in the model. The Evaluators for each Issue must allow for this and be "conservative". Another ramification of this principle is that the system should be cautious when tutoring an Issue that the student has recently been advised on.

Even if the diagnostic process can guarantee the weakness of an Issue at a given moment, the absence of a *good* Example of that Issue should prevent the Coach from breaking in. Thus one of the tutoring principles for enhancing a student's likeliness to remember what is said is to determine what a "good example" is:

> Principle 2: When illustrating an Issue, only use an Example (an alternative move) in which the result or outcome of that move is dramatically superior to the move made by the student.

Another basic principle that increases the chance of remembering the criticism of the Coach is to have the student episodically encode the example.

> Principle 3: After giving the student advice, permit him to incorporate the Issue immediately by allowing him to repeat his turn.

This principle not only provides him with the opportunity to observe the results of making a new move based on this Issue but is also apt to decrease his antagonism to the advice.

The final principle of this category presupposes that the student is a bit competitive and that he is less receptive to advice when he is about to lose (even if he incorporated the advice when repeating his turn).

> Principle 4: If a student is about to lose interrupt and tutor him only with moves that will keep him from losing.

INTEREST

In an informal learning situation, the student's interest stems primarily from the situation itself. A student plays a game because he enjoys it. Hence, one of the most important constraints of the Coach is not to destroy the student's inherent interest in the game by butting in too often. It would be much easier to implement a Coach that broke in whenever the student made a suboptimal move and told the student the better move. But faced with such a tutoring strategy, the student would quickly lose all interest in playing the game—especially if he were a poor player who could profit from judicious advice. Below are some of the principles incorporated into WEST to prevent it from being oppressive. The first two principles are the most obvious:

> Principle 5: Do not tutor on two consecutive moves, no matter what.
> Principle 6: Do not tutor before the student has a chance to discover the game for himself.

When a new student first sits down to play the game or when a student who has not played in a while returns to the game, he will take some time to familiarize himself with its mechanics. He will be using cognitive resources to figure out, for example, how to type in an expression. It is unreasonable to expect him to perform at his best when it comes to actually choosing a move before he feels fairly comfortable with the mechanics of the game.

> Principle 7: Do not provide only criticism when the Tutor breaks in! If the student makes an exceptional move, identify why it is good and congratulate him.

In WEST this is done whenever a FAIR player makes an optimal move or whenever a player makes an optimal move that uses an Issue in which he is weak. Note the various uses of the Expert just to carry out this one principle.

This next principle has appeared before in a slightly different form.

> Principle 8: After giving advice to the student, offer him a chance to retake his turn, but do not force him to.

If the student can use the Tutor's advice to improve his position in the game, he may be more attentive, but he should be given a chance to refuse to retake his turn, since he may consider a retake to be a subtle form of cheating.†

INCREASING THE CHANCES OF LEARNING

The next two principles were designed to increase the chances of learning from the gaming environment independent of the Coach's comments on the progress of the game.

> Principle 9: Always have the Computer Expert play an optimal game.

The student should be able to observe and learn from the best possible play of his opponent (typically the computer). One of the best metaskills that a student can learn from WEST (or any game) is to watch what your opponent is doing, especially if you are losing. To maximize the chance of the student seeing the value of this heuristic, he

† If WEST is being used in the mode where two students are playing against each other, the ability to retake turns after advice is turned off.

should always have a chance to observe expert play. Also, if the student realizes that the computer is not playing the best possible game, he may feel that he is being played down to and consequently lose interest in playing.

Principle 10: If the student asks for help, provide several levels of hints.

In WEST there are four levels of help. The first request for help causes the Coach to look at the student model for his current weaknesses. If a weakness is found in a skill that is required for an optimal move at this point in the game, the student is told to consider that Issue. For example, if the student is weak on the PARENTHESES issue and the optimal move for this turn requires parentheses, the student will be told "Why don't you try to use parentheses to change the order in which operations are done." The second request for help on the same move provides the student with the set of possible outcomes. For the third request, the Coach will select the outcome that it considers best. The fourth request causes the Coach to give the student an arithmetic expression that brings about the best outcome. Thus, the four successive levels of hints are based on the following rules:

Hint 1: Isolate a weakness and directly address that weakness.
Hint 2 (what): Delineate what the space of possible moves is at this point in the game.
Hint 3 (why): Select the optimal move and tell him why it's optimal.
Hint 4 (how): Describe how to make that optimal move.

ENVIRONMENTAL CONSIDERATIONS

While most of the interest in a gaming environment is derived from the game itself, many things can be done to the environment to make it more interesting. Graphics is a prime example. Playing against the computer is another. (Many CAI games have survived solely on the basis of these two considerations.) In this section we discuss some more subtle considerations that WEST employs. The next principle attempts to keep the student from getting discouraged.

Principle 11: If the student is losing consistently, adjust the level of play.

Notice that this principle conflicts with an earlier principle of always having the computer play an optimal game so that the student will have a model of expert play. For games in which there are several levels of structure to the play, such as chess, it may be better for the student to have a role model (hence opponent) which is only slightly above his level. This will tend to keep the games close while still providing examples of better moves. Our solution of this conflict is to give the computer bad spinners when it is ahead by an amount that varies with the quality of the player.

Principle 12: If the student makes a potentially careless error, be forgiving. But provide explicit commentary in case it was not just careless.

The system should be friendly about a student's error that may be from misinterpreting the rules of the game or from mistyping a move. On such errors, the system should not only allow the student to correct his mistake but, if a general rule of the game has been violated, it should draw attention to the rule and provide specific instances of it that are legal. For example, the WEST system has compiled into it diagnostic routines for many typical errors that a student is apt to make (such as precedence errors in arithmetic and giving as the value of his expression the end position of the move).

Although the twelve principles listed here are compiled into our system, it is our hope that at some future time these principles can be directly interpreted from a declarative representation of them. Such a representation could provide a *meta-environment* in which student teachers could modify and extend the rules and witness the effects on students. (See O'Shea's article in this issue for a further discussion on this point.) In WEST, a small advance along this dimension has been made by enabling the Coach to articulate all the pros and cons of what it should do next. Of course, the Coach's cogitation is not part of what a player sees as he is playing the game but instead is displayed on a second "screen". This trace of the Coach's behavior provides a graphic illustration of how many of the above principles interact to produce some very subtle tutorial behavior.

Analysis of modeling methodology

Thus far we have provided a glimpse into the underlying principles of our Coaching system as well as a simplified description of how a differential diagnostic model can be inferred from a student's behavior. It should now be clear how important the diagnostic model is to the successful execution of the top-level Issues and Examples Coaching paradigm. Consequently, we feel it is important to examine some of the limitations and underlying problems of this scheme that have not yet been discussed. We will begin with a more formal examination of the modeling process.

The inputs to the Modeler are the student's move and the set of better moves that the student could have made. Each of these moves has associated with it a set of requisite "Issues," which must be employed (in some manner) to obtain that move. For example, if the move M was to go back 2 spaces to land on a shortcut, the Issues of SHORTCUT, SUBTRACTION and BACKWARD are all required. From the student move, the Modeler can infer that the student knows the Issues needed for that move.†

What can be gleaned from knowing the set of *better moves* that the student did *not* take? In general, for each better move M, we only know that at least *one* of the set of Issues required for M was not employed and therefore reflects a potential weakness on the part of the student. But how do we know which of these Issues blocked the student from making that move? This is what we refer to as the "apportionment of blame/credit" problem: How should the Modeler apportion blame among the requisite Issues for the student's failure to discover a move?

Our solution in WEST has been to apportion blame more or less equally among all of the Issues required for the missed better moves.‡ One effect of this decision is the

† Even this cannot be inferred if there is more than one way to derive the move and the "Issues" deal with derivational rules. In WEST, the Issues are all things that uniquely underlie or are manifest in a move.

‡ In case the Modeler has more than one move that is better than the one the student made, it would be possible to find the intersection of the Issues required for each move. Unfortunately, the student is, in general, weak on more than one Issue, so this intersection will often be empty, meaning that at least two of the better moves were blocked for independent reasons. Since the evaluators have to work with noise in any case, we did not include this noise reduction heuristic. It has not proven to be a difficulty. The Coach does use this strategy when selecting an Issue to tutor. If one Issue is needed for all better moves, it is selected as the one most likely to have been missed.

introduction of incorrect information or "noise" into the model. That is, blame will almost certainly be apportioned to Issues that are in fact understood.

Having to overcome this source of noise is an excellent example of how diagnosing a student in a problem-solving situation in which the student is in total *control* is inherently more problematic than the standard mixed-initiative instructional system. In mixed-initiative systems, the Modeler can always construct a differential hypothesis from this source of ambiguity, pose a task to the student, and see what he does. Because it can create a sequence of such tasks, each one eliminating contending hypotheses, the Modeler can converge on the actual afflicting weaknesses. However, such intrusions by the Modeler into the gaming or problem-solving matrix could destroy the concentration and goal directedness of the student—creating an antidote potentially more destructive than the *raison-d'être* for a student model in the first place.

The simplified view of a student's move as a set of issues that somehow underlies the generation of the move suggests several other areas of concern in the modeling process. Since the system does not have a complete glass-box Expert, (does not account for the *entire* process that a person would use to derive the move) the set of Issues does not necessarily account for everything required to derive the move. This opens up the possibility that the underlying reason the student didn't make a move may not be one of the known Issues at all, but might instead be some other skill that has not been articulated as an Issue.† Any incompleteness in the set of Issues results in more noise in the differential student model.

An additional source of noise in the model is that students are seldom completely consistent. They often forget to use techniques that they know or get tired and accept a move that is easy to generate.

Another source of noise is learning. As the student plays the game, we hope he will be acquiring new skills that previously would have shown up as weaknesses. Even after a student learns an Issue, his model will continue to show the weakness that has accumulated over time. Ideally, the old pieces of the model should decay with time. Unfortunately, the costs involved in this computation are prohibitive. To avoid this failing of the model, the WEST Coach removes from consideration any Issues that the student has used recently (in the last three moves).

To combat the noise which arises in the model, the Evaluator for each Issue is implemented as a separate procedure. This allows individual tuning of the Evaluators in response to perceived failings. In WEST, the Evaluators use a comparison of the "taken fields" of the model with the "missed fields." The comparison percentages are adjusted to be high enough to yield conservative Evaluators. This alleviates the problems that might be caused by noise for less conservative techniques. Some coaching opportunities may be missed but eventually if the student has a problem addressed by an Issue a pattern will emerge.

STRATEGIES VERSUS ISSUES

In the scheme discussed above, the Expert is used to create a list of better moves, and then the Modeler diagnoses the student's weakness on the assumption that he did not make any of these better moves because he had not mastered one of the requisite skills

† If the Coach does not have an Issue, it will not break in, because the student's weakness may be beyond its scope. For this reason, the Issues define the space of weaknesses the Coach will try to correct.

or Issues underlying them. But what happens if the student is employing a strategy different from the Expert's? In such cases, the reason a student did not make a particular better move might simply be that he did not *want* to make it. According to his strategy, his move was the best one possible.

In order to cope with this problem, the Modeler must be able to detect when the student is using some other strategy and to characterize precisely what this other strategy is. If an executable description of the alternative strategy can be formed, then the Expert can be modified to use the new strategy. The Modeler can then reconstruct the differential student model on the basis of the modified Expert in order to separate out what Issues (as opposed to strategies) the student is weak on. Each of these tasks has its own complications. Let us proceed in this discussion under the simplifying assumption that the student maintains a consistent strategy and a consistent set of weaknesses during the period over which the model is being created.†

DIAGNOSING THE EXISTENCE OF A POSSIBLE ALTERNATIVE STRATEGY

If a modeling scheme looks at only one move of a student, it is impossible for it to determine whether the student's failure to make another move stemmed from a lack of a given skill or from harboring a suboptimal strategy.‡ However, from a *sequence* of student moves it may be possible to make such a separation. This results from the assumption that the student's strategy remains the same over the sequence of moves, whereas the Issues are likely to change from one move to the next.

The technique for detecting when a student is using a strategy different from the Expert's involves the amount of "tear" in the student model. Briefly, tear is a measure of the consistency of use of Issues. Tear starts to develop when several issues begin to reflect both a substantial amount of use when they should not have been used and non-use when they should have been. If tear in a model gets large enough, the Modeler is willing to expend some effort in conjecturing alternative strategies that the student might be using. Any alternative strategies can then be tested by re-running the Modeler over the student's past moves and comparing his behavior to that of the Expert using the conjectured strategy. If the resulting model has substantially less tear, then the conjectured strategy is taken to be a more accurate approximation of the student's strategy and is used to form the differential model. If the resulting model is not substantially more consistent, then this alternative strategy is rejected and other conjectures are tried until all reasonable conjectures are tested. Of course, for this classical "generate and test" heuristic to work, not only must the Modeler be able to generate reasonable alternative strategies, but the Expert also must be able to simulate the strategies (the conjectures must be runnable by the Expert) in order to be able to reconstruct and test the resulting student model.

CONJECTURING ALTERNATIVE STRATEGIES

Conjecturing alternative strategies is extremely difficult unless one has a sufficiently closed world that the set of possible strategies can be characterized. This

† A typical period is usually one session of play, consisting of a couple of games. Longer periods require a partitioning or layering of the model to capture the change or growth of a student's knowledge.

‡ Here again, we continue with the assumption that the Modeler is a watcher and not a manipulator of the environment and hence cannot interrupt the activity and pose its own task.

characterization can take the form of either a generative mechanism (e.g. a grammar) that synthesizes the alternative strategies (Miller & Goldstein, 1977, and also see Miller's article in this issue), or an explicit enumeration of possible alternative strategies. The world of WEST is sufficiently closed and small enough that the latter technique appears to work.

WEST's alternative strategies fall into two categories—those that are suboptimal because of a "mind bug" about the structure of the game and those that reflect an alteration in the spirit or rules of the game. An example of a "mind bug" would occur when a student always tries to move as far ahead as is possible given the particular spinner values—a nearly optimal strategy but one that overlooks the potential value of bumping your opponent. An example of an alteration of the spirit of the game occurs when the student is obsessed with bumping his opponent (e.g. because of the pretty graphics effect) and will always bump whenever a chance arises. Another example that reflects the subtlety of this category is the student who becomes fixated on getting the Coach to "speak" or interact with him. This student no longer cares about winning the game but instead becomes involved in psyching out the actual teaching strategies embedded in the system—an extremely interesting "meta-game". It should be remembered that the Coach is very conservative and will not break into the student's game unless there is a consistent pattern of poor behavior that the Coach can address. If the student is doing something completely "off the wall" it is unlikely that the Coach will break in.

Once a grammar or an explicit list of alternative strategies is created, one may determine the set of alternative strategies that a player may be using by creating a "handle" or feature recognizer (similar to an issue recognizer) for each strategy (or grammar rule).† Then, as the Modeler is accruing evidence for perceived student weaknesses on Issues, it can also be accruing evidence on possible alternative strategies by seeing which strategy features are present in each move. These features act solely as a heuristic. They are seldom unique to a given strategy, as several alternative strategies are likely to be consistent with any one move. For example, the strategy of making a maximal number might produce the same move as the strategy of maximizing the distance ahead of your opponent.

In summary, these strategy features provide *local* evidence about what alternative strategies the student may be using. A strategy for which there is local evidence is then used by the Modeler to construct a new hypothetical differential model. This new model provides a *global* check on the strategy by determining how much the tear of the differential model has been reduced.

In order to test the diagnostic sensitivity of this technique to distinguish actual student weaknesses from alternative student strategies, we have constructed various automated students (an idea proposed in Goldstein, 1977) that play with specific weaknesses *and* simultaneously with alternative strategies. These tests indicate that the technique just described is effective for WEST. We fully recognize the limited nature of this problem for the WEST "world" and are cautious in our belief that these techniques will suffice for more complex worlds.

† Such feature recognizers can be quite complex and often require properties of the space of possible moves instead of just the given student move. For example, one feature might concern whether the move involved the maximum *possible* number given the particular spinners.

Experiences with WEST

The basic Coaching system was completed in Spring of 1975 (Burton & Brown, 1976). At that time, we ran an informal experiment with 18 student teachers, in which each one used the system for at least one hour. Afterward, each was asked to complete a questionnaire about the Coach's performance. All but one had received advice from the Coach. Nine of the teachers commented favorably about the Coach's advice. Two others disagreed; one said that the Coach was offering a strategy that he did not feel he should follow because it would leave him "vulnerable to attack," an element of strategy not known to the Expert. Eight of ten subjects found the comments helpful in learning a better way to play the game and, most important, nine out of ten felt that the *Coach manifested a good understanding of their weaknesses*! One subject commented, "I misunderstood a rule; the computer picked it up in the second game."

WEST has also been used in elementary school classrooms. In a controlled experiment, the coached version of WEST was compared to an uncoached version. Table 1 gives the distribution of move patterns for the coached and uncoached groups. The

TABLE 1
Comparison between coached and uncoached groups of the percentage of times each move pattern was used when it was the best move

Pattern	Coached group (%)†	Control group (%)†
(A+B)−C	72	74
(A*B)+C	57	58
(A*B)−C	41	46
(A+B)*C	65	44
A−(B+C)	13	29
A*(B−C)	32	22
(A*B)/C	23	9
A/(B−C)	25	0
A−(B/C)	14	0
(A/B)−C	14	0
(A−B)/C	14	0
A−(B*C)	13	0
(A+B)/C	0	0
A/(B*C)	0	0
A/(B+C)	0	0
A+(B/C)	0	0

Special moves

Control group (%)†		Coached group (%)†	
TOWN:	72	TOWN:	79
BUMP:	18	BUMP:	54
SHORCUT:	41	SHORTCUT:	54

† % of time pattern was taken and was best.

coached students showed a considerably greater variety of patterns, indicating that they had acquired many of the more subtle patterns and had not fallen permanently into "ruts" that prevented them from seeing the relatively rare occasions when such moves were important. Probably the most surprising result from this experiment was that the students in the coached group enjoyed playing the game considerably more than the uncoached group. This finding was especially significant, because one of our greatest fears had been that our coaching principles were sufficiently ill-developed that either the Coach would interrupt too often, destroying the inherent enjoyment of the game or too seldom, failing to get students out of ruts. We have not yet had the opportunity to explore why, in fact, students seem to prefer the game with the Coach. One interesting hypothesis is that the students using the Coaching version were actually engaged in a meta-game of "psyching out" the Coach to get it to speak. If this rather romantic hypothesis turns out to be valid, it would open a new arena for conveying some of the very important survival principles for formal education.

References

BROWN, J. S. & BURTON, R. R. (1975). Multiple representations of knowledge for tutorial reasoning. In BOBROW, D. & COLLINS, A. Eds, *Representation and Understanding: Studies in Cognitive Science*. New York: Academic Press.

BROWN, J. S. & BURTON, R. R. (1978). Diagnostic models for procedural bugs in basic mathematical skills. *Cognitive Science*, **2**, 155–192.

BURTON, R. R. & BROWN, J. S. (1976). A tutoring and student modeling paradigm for gaming environments. In *Proceedings for the Symposium on Computer Science and Education*, February.

CARR, B. & GOLDSTEIN, I. (1977). Overlays: A theory of modeling for computer aided instruction. *Artificial Intelligence Memo 406*. Massachusetts Institute of Technology, Cambridge, Massachusetts.

CLANCEY, W. (1979). Tutoring rules for guiding a case method dialogue. *International Journal of Man–Machine Studies*, **11**, 25–49.

DUGDALE, S. & KIBBEY, D. (1977). *Elementary Mathematics with PLATO*. Urbana, Ill.: University of Illinois (Computer-based Education Research Laboratory), July.

FISCHER, G., BROWN, J. S. & BURTON, R. R. (1978). Aspects of a theory of simplification, debugging, and coaching. In *Proceedings of the Second Annual Conference of Canadian Society for Computational Studies of Intelligence*, July.

GOLDSTEIN, I. P. (1977). The computer as coach: an athletic paradigm for intellectual education. *Artificial Intelligence Memo 389*. Massachusetts Institute of Technology, Cambridge, Massachusetts.

GOLDSTEIN, I. P. (1979). The genetic graph: a representation for the evolution of procedural knowledge. *International Journal of Man–Machine Studies*, **11**, 51–77.

GOLDSTEIN, I. & PAPERT, S. (1977). Artificial Intelligence, language, and the study of knowledge. *Cognitive Science*, **1**, (1), 1–21.

GROEN, G. J. (1978). The theoretical ideas of Piaget and educational practice. To appear in SUPPES, P. Ed., *Impact of Research on Education: Some Case Studies*. Washington, D.C.: National Academy of Education.

MILLER, M. L. (1979). A structured planning and debugging environment for elementary programming. *International Journal of Man–Machine Studies*, **11**, 79–95.

MILLER, M. L. & GOLDSTEIN, I. (1977). Problem solving grammars as formal tools for Intelligent CAI. In *Proceedings of Association of Computing Machinery*.

O'SHEA, T. (1979). A self-improving quadratic tutor. *International Journal of Man–Machine Studies*, **11**, 97–124.

SLEEMAN, D. H. & HENDLEY, R. J. ACE: a system which Analyses Complex Explanations. *International Journal of Man–Machine Studies*, **11**, 125–144.