DOCUMENT RESUME

ED 196 710 SE 033 691

AUTHOR	Goldstein, Ira
TITLE	A Research Study of Computer-Based Tutoring of Mathematical and Scientific Knowledge. Final Technical Report.
INSTITUTION	Massachusetts Inst. of Tech., Cambridge. Artificial Intelligence Lab.
SPONS AGENCY	National Science Foundation, Washington, D.C.
PUB DATE	80
GRANT	NSF-SED77-19279
NOTE	36p.
EDRS PRICE	MF01/PC02 Plus Postage.
DESCRIPTORS	*Artificial Intelligence; Cognitive Processes; *Computer Assisted Instruction; *Educational Games; Logic; *Mathematics Education; *Mathematics Instruction; *Problem Solving; Teaching Skills
IDENTIFIERS	*Mathematics Education Research

ABSTRACT

Computer coaching of students as an aid in problem-solving instruction is discussed. This report describes an advanced form of computer-assisted instruction that must not only present the material to be taught, but also analyze the student's responses. The program must decide whether to intervene and how much to say to a pupil based on its knowledge of the player or players in an ever-changing context. The focus of this project is problem solving using the Wumpus Game. An evolution of computer coaches from a simple rule-based approach to a more complex model is described. (MP)

Massachusetts Institute of Technology

Artificial Intelligence Laboratory

Final Technical Report

NSF Grant SED77-19279

A research study of computer-based tutoring of mathematical and scientific knowledge[#]

Ira Goldstein[$]

Abstract

This report describes the evolution of a problem solving model over several generations of *computer coaches*. Computer coaching is a type of computer assisted instruction in which the coaching program observes the performance of a student engaged in some intellectual game. The coach's function is to intervene occasionally in student generated situations to discuss appropriate skills that might improve the student's play. Coaching is a natural context in which to investigate the teaching and learning processes, but it is a demanding task. The computer must be able to analyze the student's performance in terms of a model of the underlying problem solving skills. This model must represent not only expertise for the task but also intermediate stages of problem solving skill and typical difficulties encountered by the learner. Implementing several generations of computer coaches to meet these demands has resulted in a model that represents problem solving skills as an *evolving set of rules* for a domain acting on an *evolving representation of the problem* and executed by a *resource-limited problem solver*. This report describes this evolution from its starting point as a simple rule-based approach to its current form.

Keywords: Information processing psychology, artificial intelligence, computer assisted instruction, cognitive science, knowledge representation, computer games.

This report was published in Reif, F. and D. T. Tuma (eds.), *Problem Solving and Education: Issues in Teaching and Research*, Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1980, pp. 53-80.

$ The author is now at the Xerox Palo Alto Research Center, 3333 Coyote Hill Road, Palo Alto, Cal. 94304.

Table of Contents

Computer coaching is an advanced form of Computer Assisted Instruction. 4

The Wumpus Game has served as our experimental testbed. 5

Our first step was to represent problem solving skills as a set of rules. 8

Our second step was to represent problem solving skills as an evolving set of rules. 11

- Extending the coach's representation of problem solving skills improves its student modelling capabilities.
- A Student Simulator was implemented to explore the behavior of different skill models.

Our third step was to represent the associated data structures. 15

- A Display-based Wumpus game was implemented to explore the role of data representation.
- We are now in the process of implementing a coach for Display Wumpus.

Our fourth step was to represent the resource limitations of the problem solver. 18

- An elementary method for estimating the complexity of the game was implemented.
- A theory-based estimate of cognitive load is needed.

There are more steps to be taken in constructing an adequate problem solving model. 21

A computational theory of problem solving has many educational applications. 23

References 24

Figures 26

For several years, we have employed an unusual methodology to study problem solving skills.*
Rather than observing students interacting with *human teachers*, we have observed their interactions with *computer coaches*. This has provided us with a controlled environment for studying teaching and learning. Our evaluation measure has been the adequacy of a problem solving theory to support successful tutoring by the coaching program.

* This paper describes research done with my students and colleagues at MIT, especially Jim Stansfield, Brian Carr, Kurt van Lehn, Barbara White, David Ihrie and Sandor Schoichet. It has also benefited from a close collaboration with John Seely Brown and his colleagues at Bolt, Beranek and Newman.

Developing a computational representation for problem solving skills is obviously difficult. Hence, our strategy has been to develop this representation in an incremental way.

The first step introduced a representation for expertise as a set of rules. This gave the coach a description of the goal state of the teaching process.

The second step added a representation for the *evolution of these rules* in which different levels of skill were described explicitly.

The third step added a representation for the *data structures* employed by the student, since his problem solving skills clearly include strategies for structuring the problem.

The fourth step added a representation for the *cognitive resources* of the student, since otherwise the coach could not distinguish between problems requiring the same skills but making different demands on memory and processing power.

Before recounting this evolution, it is useful to observe that our research differs significantly from traditional studies of problem solving typified by Polya [1957]. Polya concentrates on enumerating useful heuristics. Our research is complementary, being less concerned with problem solving heuristics than with arriving at a general representation for such heuristics within the overall problem solving context. Hence, many of our rules are domain specific. But the overall architecture -- a processsor applying rules, manipulating data structures, and constructing new rules from old -- is general. In a sense, we have focussed more on the *form* than on the *content* of problem solving. However, deriving a better understanding of the architecture of problem solving is crucial if we are to embed the presentation of particular problem solving skills within a more comprehensible framework. It is also crucial if we are to develop an improved educational technology based on computers.

Computer coaching is an advanced form of Computer Assisted Instruction.

In traditional computer assisted instruction, the computer's understanding is minimal. Generally, the material is represented as a script, and the computer's function is to direct the presentation of material based on keyword responses to pre-programmed queries. Computer coaching, however, does not allow such cookbook methods. The coach must advise players in a constantly changing context. To meet this objective, coaching systems have the structure shown in figure 1.

The *Expert* module generates solutions to the student's problem. Within the game context, these solutions are analyses of the pros and cons of alternative moves. To formulate these analyses, the Expert uses a procedural representation of problem solving skills for the domain. Thus, the design of the expert requires a formal and complete study of the knowledge demanded by the task.

The *Psychologist* module must compare the Expert's analysis with the student's performance to hypothesize which skills the student understands. These hypotheses are stored in an Overlay Model, a term I use to emphasize that the model is defined in terms of the coach's overall understanding of the problem domain. Inferring this model is a difficult task for human teachers. Hence, if the computer coach is to succeed in its restricted world, it must take recognizance of as many sources of evidence as we can make available to it. Therefore, the design of the Psychologist focusses on developing programs to examine the student's play, ask occasional questions, request a background questionnaire and assess the intrinsic complexity of material in the syllabus.

The *Tutor* module is alerted by the Psychologist to situations in which the student has not employed a skill and hence made a less than optimal move. The Tutor must then decide whether to intervene and how much to say. This decision is made by employing a set of teaching heuristics. Hence, the design of the Tutor directly raises both educational questions related to the nature of explanations and linguistic questions related to the expression of these explanations in English.

Thus, coaching systems are complex, requiring a representation both of the skills to be taught, and of the procedures by which modelling and tutoring can be accomplished. Our coaching

programs reflect this complexity both in their size (several hundred thousand words of code) and their development time (typically several man years). To date, coaching experiments have been conducted for a limited, but interesting range of domains including geography [Carbonell, 1970], electronic troubleshooting [Brown, Burton & Bell, 1975], nuclear magnetic resonance spectra analysis [Sleeman, 1975], medical diagnosis [Clancey, 1978], programming [Miller, 1978], and mathematical games [Burton & Brown, 1976; Goldstein & Carr, 1977].

In this paper, however, I shall eschew the broad view by applying a microscope to the central box of figure 1. Developing a satisfactory representation for problem solving skills is clearly essential to the design of coaches, as the centrality of the box in the diagram indicates.

The Wumpus Game has served as our experimental testbed.

In 1975, we began an examination of procedural models for problem solving in a game environment. We choose a game environment based on Burton and Brown's [1978] experience with a coach for the Plato project's arithmetic game *How the West was Won**. They found games to be a motivating but nevertheless constrained environment that was well suited to the coaching paradigm.

* WEST is a simple board game in which the player moves by forming an arithmetic expression from three spinners. The game is a race and hence the player's usual goal is to form the largest expression, with the typical alternatives of bumping your opponent or landing on special squares. The student gains experience with arithmetic by searching for the optimal expression. The coach tracks the student's choice of move and offers advice about arithmetic skills not being employed by the student in situations where their use would be of value.

Following Burton and Brown's lead, we designed a coach for Wumpus, a computer game invented by Gregory Yob [1975]. The game is a modern day *Theseus and the Minotaur* in which the player's goal is to slay the Wumpus. The game's virtue is that an interesting variety of logical and probabilistic reasoning skills are required to play skillfully. To define the game, I have reproduced the introduction printed by the coach.

> *You are a world-renowned hunter descending down into the caves of darkness, lair of the infamous man-eating Wumpus. To win the game, you must kill the Wumpus by shooting one of your five arrows into his lair from a neighboring cave. If you go into the cave of the Wumpus he will eat you. Within the warren there are two other kinds of dangers, bats and pits. The pits are bottomless and fatal if you fall into of them. If you visit the home cave of bats they will pick you up and carry you to*

another cave which might contain the Wumpus or a pit (either of which is fatal).

You can gain information about the warren through exploration. If bats are in one of the neighboring caves you will hear their high-pitched squeaking. Likewise, if you are next to a cave with a pit you will feel a chilling draft. If you are near the Wumpus, you will smell his horrible stench.

Before you shoot an arrow, you should consider the fact that if the Wumpus is not in the cave, the arrow will start ricocheting and may kill you (and it is an agonizing death). If it does start ricocheting, there is roughly a one third chance that it will kill you. If you run out of arrows without slaying the Wumpus, you are doomed to wander forever until you die.

Your only companion in this endeavor is the wise old sage, Wusor. If you make a hasty move, Wusor may stop you and give you advice, but the final decision rests with you.

Following this introduction, a typical game might proceed as follows. The student is Mary, a 12 year old player.

Hi, Mary. There are 15 caves, 2 bats, 2 pits and 1 Wumpus. You are at cave 15 with neighbors 4, 14 and 0. Brrr! There is a draft. You are near a pit. What now?

> 4

You are at cave 4 with neighbors 15, 14 and 2. Brrr! There is a draft. Squeak! A bat is near.

As figure 2 illustrates, many players draw maps and/or tables to keep track of their exploration of the warren. Figure 2 shows the map drawn by Mary for this purpose as well as alternative representations employed by other students.

Wumpus is colorful and challenging, thereby capturing the interest of a wide range of players. Adults typically invest several hours to master the game: children several days or weeks. Locating multiple dangers in a randomly connected warren requires skills of logical deduction, probabilistic inference, problem representation, and overall strategy selection. For example, consider Mary's situation.

From a logical standpoint, Mary should infer that cave 2 contains a bat since (1) the *squeak* in cave 4 implies that a bat is in either cave 2 or 14, but (2) the absence of a squeak in 15 rules out 14 as a possibility. As Mary's game progresses, opportunities will be commonplace for arguments by elimination, by cases or by contradiction.

From a probabilistic standpoint, Mary should infer that cave 14 is more dangerous with respect to pits than 0 or 2 on the basis of the multiple warnings. The multiple warnings for a pit do not determine the location of the danger. (If there were only one pit, this would be not be true.) But multiple warnings do imply that cave 14 should be treated as more likely to contain the pit. An expert player typically makes approximate numerical judgments of the probabilities when logical inferences are insufficient to locate safe caves.

From a problem representation standpoint, Mary's map is a useful artifact for representing the problem. Many students initially choose tables, which make deductions about connectivity difficult. There are, however, other representational devices that prove useful such as lists or tables. The most challenging aspect of the game for many players is to derive an adequate representational scheme.

From a strategic standpoint, Mary must recognize that her goal is to avoid the more dangerous caves while still gaining information about the warren. Strategic considerations grow more complex as the number of arrows are exhausted or the time to complete the game grows short.

This analysis only sketches the requisite knowledge, but it demonstrates that skilled play does pose an intellectual challenge. Indeed, the game is sufficiently complex to exhibit a *plateau phenomenon* in which players occasionally stagnate at particular levels of skill. Tutoring is then required to facilitate further learning. Hence, the game is not an artificial environment in which to study problem solving.

Our first step was to represent problem solving skills as a set of rules.

In 1976 we implemented a coach in which the mathematical and probabilistic skills needed to play Wumpus were represented by approximately 25 rules [Goldstein & Carr, 1977]. The rules fell into two categories: those that deduced evidence about the warren and those that made strategic decisions on the basis of that evidence. A few are illustrated below:

<u>Typical Evidence Rules</u>: (These rules construct sets of caves that represent hypotheses about the locations of different dangers. The context for all of these rules is that the player has just entered a cave in the warren and been told its neighbors and its warnings.)

ER1: *Add the unvisited neighbors to the set of FRINGE caves. The FRINGE set records those caves that have not yet been visited but can be reached from the player's current location in the warren.*

ER2P: *If there is not a pit warning, then add the neighbors to PIT-. PIT- is the set of caves that do* <u>*not*</u> *risk pits.*

ER3P: *If there is a pit warning, then add the neighbors to PIT+. PIT+ is the set of caves that risk pits.*

ER5P: *If there is a pit warning and one of the neighbors is already in PIT+, then add that neighbor to PIT2. PIT2 is the set of caves for which the player has multiple evidence of a pit.*

ER6P: *If there is a pit warning, and all but one of the neighbors are known to be safe, then add that neighbor to PIT=. PIT= is the set of caves that definitely contain a pit.*

Similar rules are defined for the other dangers. The convention is employed that special case rules for particular dangers are given names ending with the suffixes B, P or W.

<u>Typical Strategy Rules</u>: (These rules are concerned with choosing the move based on the available evidence.)

SR0: *Shoot an arrow if the Wumpus' lair is found, that is, shoot if a cave is added to WUM=.*

SR1: Explore safe caves, that is, explore any caves in the intersection of PIT-, BAT- *and* WUM-.

SR2: Explore caves implicated by single warnings before caves implicated by multiple warnings, that is, if possible, prefer caves that are not in DANGER2 *sets.*

SR3: Explore caves that only risk bats, that is, prefer elements of BAT+.

SR4: If no other strategy rule applies, explore any available fringe cave, i.e. any member of FRINGE.

Similar strategy rules express preferences for other combinations of risk as recorded in the various evidence sets.

These rules were applied in a fixed order, with the strategy rules being more sensitive to the ordering chosen than the evidence rules.

These rules are problem specific. Hence the question arises whether or not the coach conveys any general problem solving knowledge. My reply is that these rules implicitly embody important general skills such as (1) argument by elimination, (2) the use of sets to represent hypotheses, and (3) the sequential organization of a set of heuristics. It is true that the coach does not understand them in this fashion: its orientation is problem specific. However, a reasonable hypothesis is that general problem solving skills can be profitably taught through problem specific situations. This is not a radical position since it accords with the traditional philosophy of *learning by doing*. Eventually, we will conduct experiments to learn if the implicit skills of Wumpus do transfer to other situations. But our current focus is on the prerequisite goal of developing an adequate representational formalism to incorporate the required problem solving knowledge into a coach.

Within the computer, these rules are represented as a set of attribute value pairs. The description for ER3P is shown below:

DESCRIPTION OF RULE ER3P

ATTRIBUTE	*VALUE*
TYPE	EVIDENCE-RULE
ENGLISH	IF A CAVE HAS A DRAFT, THEN THE NEIGHBORS POSSIBLY RISK PITS.
CONDITION	IS "DRAFT" A WARNING OF THE CURRENT CAVE?
ACTION	ADD THE NEIGHBORS OF THE CURRENT CAVE TO PIT+.

The rules are represented in this form, rather than as lines of code in a large program, for two reasons. First, the coach must diagnose situations where the student knows the individual rules but applies them in the wrong order. Second, the coach must know facts for explaining the rule that are not essential for their *execution*. Indeed, this second point really encapsulates the theme of this paper. Our first rule-oriented, representation for problem solving skills was adequate to define an expert program, but was insufficient to guide a teaching program. Our three extensions addressed this limitation by supplying knowledge about problem solving skills that a teacher requires to explain them successfully.

Based on these rules, the coach was able to offer the following kind of advice. Mary has just chosen to move to cave 14 in the situation illustrated in figure 2:

> ***Explanation E1:*** *Mary, it isn't necessary to take such large risks with pits. Multiple evidence is more dangerous than single evidence for pits. Here there is multiple evidence for a pit in 14 and single evidence for a pit in 0. Perhaps we should explore cave 0 instead. Do you want to take back your move?*

This explanation was generated by a set of templates applied to a *proof* generated by the Expert module that move 0 is better than move 14. The term *proof* is legitimate when the rules are viewed as theorems of the domain. For example, the Expert produces the following (simplified) analysis. The analysis is expressed in English rather than the equivalent internal computer representation.

There is no squeak in cave 15.	*;Given.*
Therefore, cave 0 is a member of bat-.	*;By evidence rule ER2B.*
There is a draft in cave 15.	*;Given.*
Therefore, cave 0 is a member of pit+.	*;By evidence rule ER3P.*
There is a draft in cave 15 and cave 4.	*;Given.*
Therefore, cave 14 is a member of pit2.	*;By evidence rule ER5P.*
Conclusion, 0 is superior to 14.	*;By strategy rule SR2.*

Thus, our first Coach was essentially a *mathematician* in the sense that it viewed the tutoring process from the theorem proving standpoint. Its goal was to inform the student of the bugs in his *proof* of the current situation. Our experience with this coach was that students generally enjoyed its advice. And, upon occasion, it successfully prodded students off plateaus by making them aware of poor moves. However, viewing students as mathematicians who need only be told the appropriate theorems is clearly insufficient as a model of the learning process. The next section

discusses how the addition of an historical perspective on the development of these theorems significantly improves the coach's explanatory power.

Our second step was to represent problem solving skills as an *evolving* set of rules.

To appreciate why an historical perspective of problem solving skills must supplement the coach's basic mathematical understanding, it is useful to reexamine the advice offered Mary in the previous section. Recall that she has made a poor move to cave 14. The mathematical explanation provided is that she has failed to apply the *double evidence* theorem. The central assertion was:

> ***Explanation E1:*** *Mary, it isn't necessary to take such large risks with pits. Multiple evidence is more dangerous than single evidence for pits. ...*

This advice might be sufficient. Indeed, in our experience, it is just right for some students. But the explanation does not take advantage of Mary's history. For example, if she has recently encountered a similar situation for another kind of danger, then an explanation that emphasized the analogy would be appropriate.

> ***Explanation E2:*** *Mary, it isn't necessary to take such large risks with pits. We have seen that multiple evidence is more dangerous than single evidence for bats. ...*

Or alternatively an explanation that emphasized the relevant generalization.

> ***Explanation E3:*** *Mary, it isn't necessary to take such large risks with pits. Multiple evidence is more dangerous than single evidence for all dangers.*

Finally the explanation might emphasize the relationship of the new strategy to an earlier, simplified view of the game.

> ***Explanation E4:*** *Mary, it isn't necessary to take such large risks with pits. In the past, we have distinguished between safe and dangerous evidence. Now we should distinguish between single and multiple evidence for a danger.*

I am not saying that any one explanation is the *correct* one. However, I am saying that a model of learning and teaching must make provision for these alternatives. A teacher can then move among them as the student's history and responses suggest.

The first *mathematician* coach did not have this flexibility. Its point of view was essentially that the student was an "*empty bucket*" in which knowledge was to be poured until skilled performance was achieved. At any moment in time, the coach viewed the student's skills as a subset of those of the expert. It ignored the fact that acquisition of skills is a more complex process involving the use of analogies, generalizations and corrections to build new rules from previously acquired ones. This ignorance was reflected in its inability to structure an explanation in these terms. Clearly what was needed was an extended representation for describing the evolution of procedural knowledge from one level of skill to another.

We provided this extended representation by employing a network formalism in which rules were represented as nodes and their evolutionary relationships as labelled links between these nodes. These labels designated various relationships by which one rule might be built from another, including generalization, specialization, analogy, and refinement. Figure 3 is a region of the Wumpus rule network that exhibits these relationships. Examine rule ER1 of figure 3. It is a general statement that "*If a warning occurs, then the neighbors for the current cave should be added to the set D+ representing possibly dangerous caves.*" ER3 is a generalization of particular rules for bats, pits and Wumpi. This is represented in the network by *specialization* links to ER3B, ER3P and ER3W. (There are inverse *generalization* links from ER3B, ER3P and ER3W to ER3 which are not shown.) Each of the specializations is connected to its brothers by *analogy* links. The analogy relationship is defined formally by the existence of a mapping from the variables of one special case rule to another. Finally, ER3 is connected by *refinement* links to ER4 and ER5. A refinement is defined by breaking a rule's *condition* or *action* into separate cases. Thus, ER4 and ER5 are produced by breaking the action of ER3 into two cases: one for single evidence and one for double evidence. ER3 was in turn refined from ER1 by breaking the condition of that rule into two cases: one for warnings and one for no warnings.

There is another vantage point from which to view the network. This is from the local perspective of an individual rule rather than from the global perspective of the overall network. Recall that our first representation viewed rules in isolation: their description contained only properties for their condition and action. From a local perspective, the rule network is a *derived* structure that arises from augmenting individual rule descriptions with special connections to related rules. The rule description thus characterizes rules as members of a society with both internal and external relationships. In fact, the actual computer representation takes this form. Each rule is

supplied with an augmented description, of which the one illustrated below is typical.

EXTENDED DESCRIPTION OF RULE ER3P

ATTRIBUTE	*VALUE*
COMMENT: THIS IS THE	*BASIC RULE-DESCRIPTION.*
TYPE	EVIDENCE-RULE
ENGLISH	IF A CAVE HAS A DRAFT, THEN THE NEIGHBORS POSSIBLY RISK PITS.
CONDITION	IS "DRAFT" A WARNING OF THE CURRENT CAVE?
ACTION	ADD THE NEIGHBORS OF THE CURRENT CAVE TO PIT+.
COMMENT: THESE RELATIONS	*DEFINE THE SKILL NETWORK.*
ANALOGOUS-TO	ER3B ER3W
SPECIALIZATION-OF	ER3

The network itself is derived from the connections between rules specified in the analogy, specialization and refinement attributes.

The coach cannot perform the learning processes specified by these attributes: the programmer supplied these interrelationships. However, given the availability of this network representation of the skills to be taught, the new coach now being completed will be capable of a more diverse set of explanations. The variations E2, E3 and E4 will be generated by using English templates to be triggered by the existence of evolutionary links between the specific rule that "double evidence for pits is more dangerous than single evidence" and other rules in the skill network. The region of the skill network containing this rule is shown in figure 4, embedded within an Overlay Model describing the student's knowledge state. The relationship of the network to the student model is the subject of the next section.

Extending the coach's representation of problem solving skills improves its student modelling capabilities.

In terms of modelling the student, our notion of an Overlay Model now becomes that of identifying the skill nodes employed by the student and the evolutionary links followed in the acquisition process. Figure 4 is a graphic representation of an Overlay Model maintained by the coach. The coach keeps track of which rules it believes the student already possesses on the basis of

his or her behavior as well as the explanations it has offered to facilitate this learning process.

The computer represents an Overlay model as additional information within each rule description. Thus, the description for SR2P would include the following properties:

DESCRIPTION OF RULE SR2P

ATTRIBUTE	*VALUE*	
COMMENT: THIS IS THE BASIC RULE DESCRIPTION. IT IS THIS STRATEGY RULE THAT OBJECTS TO MARY'S CHOICE OF CAVE 14 AS HER MOVE.		
TYPE	STRATEGY-RULE	
ENGLISH	IF THERE ARE CAVES THAT ONLY POSSIBLY RISK PITS, MOVE THERE.	
CONDITION	IS PIT1 A NON-EMPTY SET?	
ACTION	SET MOVES, THE SET FROM WHICH THE NEXT MOVE WILL BE CHOSEN, TO PIT1.	
COMMENT: OVERLAY MODEL INFORMATION FOR THIS RULE.		
EXPLANATIONS	E1	*;E1 IS THE EXPLANATION JUST GIVEN TO MARY REGARDING THE INADVISABILITY OF MOVING TO CAVE 14.*
USED	0	*;THE COACH BELIEVES THAT MARY HAS NEVER USED THIS RULE.*
APPROPRIATE	1	*;THE COACH BELIEVES THAT THE RULE HAS BEEN APPROPRIATE ONCE. IN THIS CASE, THIS IS IN CHOOSING NOT TO MOVE TO CAVE 14.*
FREQUENCY	0	*;"FREQUENCY" IS THE RATIO OF "USED" TO "APPROPRIATE" AND REPRESENTS AN ESTIMATE OF HOW FREQUENTLY MARY EMPLOYS THIS SKILL. FOR SR2P, THE COACH BELIEVES THAT MARY HAS NEVER EMPLOYED THIS RULE.*
KNOWS	NO	*;"KNOWS" RECORDS THE COACH'S HYPOTHESIS FOR WHETHER OR NOT MARY KNOWS THIS RULE. IT IS "NO" IF THE "FREQUENCY" IS LESS THAN .5.*

A Student Simulator was implemented to explore the behavior of different skill models.

The Student Simulator is an environment for executing the rules specified in an Overlay Model. Its function is to allow a teacher to explore the behavioral implications of different hypotheses about a student's skills. Its value arises in those situations in which the teacher is unable to predict

a priori the divergent behavior implied by different student models for complex problem solving situations.*

* [Brown & Burton, 1978, pp. 170-171] have demonstrated another utility of a student simulator, and this is as an environment in which student teachers can gain experience in building models of their pupils. They exposed student teachers to a simulator for elementary arithmetic skills. Their results showed that exposure to their simulator signficantly improved the student teacher's ability to diagnose procedural bugs in a student's behavior.

Figure 5 illustrates the simulator by showing a trace of two simulated students on the same game. The simulation in the upper half of the figure employs the rules specified by Mary's Overlay model. This rule set does not take account of double evidence, as was reflected in Mary's earlier choice of cave 14. (See figure 2.) Hence when faced with the choice of cave 8 or cave 10, it chooses cave 8, the riskier of the two. The simulator prints the explanatory message for the choice of move 6 by fetching a description of the strategy rule that governed the decision. The second simulation is constructed from the Overlay model for Jane, a more advanced player who does distinguish between single and multiple evidence as reflected in her use of BAT1 and BAT2 markers. (See figure 2 for Jane's representation of the warren.) Here the Jane simulacrum correctly chooses cave 10 as the better move. Hence, the simulator can serve a teacher who is interested in understanding the different behavior that two skill models might produce.

Our third step was to represent the external data structures employed by the student.

Supplying an historical perspective improves the range of explanations that the coach can deliver, but it is not sufficient as a *teachable* model of the problem solving process. Absent is a representation of the objects on which problem solving rules operate. The coach, as yet, has no understanding of the problems involved in formulating a representation of the problem. Being both mathematician and historian is insufficient: the coach must also be an *epistemologist*. A more careful scrutiny of the knowledge demanded by the problem solving process is in order.

Our attention was focussed on the need to consider the student's representation of the state of the game, its history, and his hypotheses about the task by the following kind of situation. Students would frequently be able to explain a skill in isolation, yet not apply it when appropriate. This point was illustrated by Thorson [1978], who conducted an experiment in which two populations of students were exposed to Wumpus with different aids to represent the game. Some were provided only pencil and paper; others a display version of the game on which a map was automatically drawn. The latter group played a much superior game of Wumpus, yet statistically both groups possessed the same skills.

Hence, it was necessary to extend our representation of problem solving to include a representation of the evolving data representations employed by the student. To guide this extension, we observed the various representations employed by students, as illustrated in figure 2. We observed that some students built data representations that reflected their ignorance of the importance of certain information: for example, Mary did not realize that distinguishing single from multiple warnings for a danger was useful. Other students assumed too much: a common bug of younger players was assuming that caves connected if they were drawn close to one another on the map. Still a third group of students created representations that proved adequate while the game was simple, but failed when its demands grew complex. Johnny's table will make inferences about the connectivity of the warren complex as the game progresses. On the other hand, a fourth class of students employed redundant representations to facilitate different inferences. For example, Jane drew both a map and a list of visited caves. She used the former to reason about connectivity and the latter to infer the location of dangers by a process of elimination.

A Display-based Wumpus game was implemented to explore the role of data representation.

To explore the role of data and hypothesis representations in problem solving, we have implemented a display-based Wumpus in which the student can manipulate different data structures. The student is not allowed pencil and paper; hence, his entire external representation is in a form that the computer coach can observe. Display Wumpus allows the student or the coach to select various data representations for the connectivity of the warren and for hypotheses formulated by the student regarding dangers from among those shown in figure 6. Here are some of the choices Display Wumpus permits:

(1) The warren can be described via the usual teletype description which prints the current caves, its neighbors and its warnings, or via a map.

(2) Caves that the player has visited can be represented either by dashing their outline or in the list labelled VISITED.

(3) The player's current location can be represented either by the cave with the face or by the last entry to the TRAIL list.

(4) An hypothesis that a cave risks a particular danger can be grouped either under the heading of the cave (on the map) or under the heading of the danger (in the tabular evidence area).

These representations are only a subset of those which a student might design on his own. We have restricted the student's freedom in return for increasing the coach's insight into his problem solving. However, Display Wumpus is only an experimental tool. To make it into a useful educational environment, Display Wumpus could be provided with a *tablet* device so that the student can design his own representation while still allowing the coach to observe his behavior.

Formally, Display Wumpus is based on the following point of view. For each type of evidence (that is, input and output variable of a rule), there is a set of possible external representations. For Wumpus, these representations include tables, maps, and the null representation, that is, no external representation at all. This selection of possible external representations is based on our experiments with children. We typically found an evolution of representation from the null representation, to tables like Johnny's, to maps like Mary's, to combined representations like Jane's. We have also seen "representation traps" in which the player persists with a given representation, say Johnny's tabular representation of the maze, and consequently finds it extremely difficult to progress to the acquisition of more complex reasoning strategies.

We are now in the process of implementing a coach for Display Wumpus.

The current Display Wumpus has served as an experimental medium for several months in which we have observed the untutored play of many students. We are now designing a coach to take advantage of the larger window for observing the student's play which it provides. The new coach will maintain an Overlay Model of the student's use of the data representations supplied by Display Wumpus. For example, consider again Mary's map of figure 2. If this map is drawn with Display Wumpus, then the coach would construct the following description to supplement its overlay model of her problem solving skills:

DESCRIPTION OF MARY'S PROBLEM REPRESENTATION

ATTRIBUTE	*VALUE*	
WARNING-REPESENTATION	MAP	*;MARY RECORDS THE DRAFT ON THE MAP*
NEIGHBOR-REPRESENTATION	MAP	*;AND THE NEIGHBORS,*
DANGER-REPRESENTATION	NONE	*;BUT NOT THE CAVES THAT RISK PITS,*
VISITED-REPRESENTATION	NONE	*;NOR THE VISITED CAVES.*

With this description, the coach will be capable of explaining a certain representation if it believes that the student knows the appropriate rules, but is not representing the data in a way that would make their application evident. For example, suppose Johnny selected cave 14 as his next move. Recall that earlier, when Mary selected this move, the coach discussed with her the increased danger implied by double evidence. But Johnny's problem may simply be that he does not realize that cave 0 is implicated both by the warning in cave 15 and the warning in cave 4. His tabular representation requires that he infer that 14 is connected to both by examining two rows of the table. It is not as easy an inference as it is for Mary with her graphic representation of the warren. In this situation, we envision the following dialog:

Explanation E5: Johnny, do you realize that cave 14 is risky both because of the warning in cave 15 and the warning in cave 4? Using tables makes this difficult to see. Perhaps you would like to employ a map?

The coach would then instruct Display Wumpus to provide a map similar to the one shown in figure 6, but simplified by removing the danger symbols (PIT1, PIT2, etc.). These symbols along with the list representation would not be offered since Johnny's level of play does not yet warrant proposing this additional machinery. Johnny could then choose to employ the map by using the appropriate drawing commands of Display Wumpus. Thus, the coach does not engage in a discussion of the probabilities implied by double evidence, but addresses the prerequisite necessity to help Johnny with his representation.

Our fourth step was to represent the resource limitations of the problem solver.

In the previous section, we provided the coach with knowledge regarding the various data structures a student might employ to represent the problem. This knowledge is necessary but not sufficient to guide the coach in its generation of explanations like E5; such knowledge does not determine *when* the corresponding advice is appropriate. For example, in the above situation, it may be premature to suggest a map to Johnny. The game may still be too simple to stress the table representation. Tutoring leverage will not exist until Johnny perceives the inadequacies of his current representation scheme relative to the complexity of the problem. Thus, whether or not advice about a change in representation is appropriate depends on an estimate of the cognitive load that the problem imposes on the student. Hence, an epistemologist's insight into the breadth of knowledge required by the task must be supplemented by a psychologist's insight into the relative

complexity of the material.

An elementary method for estimating the complexity of the game was implemented.

The coach estimates the complexity of the game in terms of the number of dangers, the number of caves, the propagation distance of warnings and the mobility of the dangers. Increasing any of these parameters is assumed to increase the complexity of the task. This is borne out by our informal experiments with students and by the intrinsic computational work that the problem solver is required to perform.

The coach currently employs its student model to guide its selection of the complexity of the Wumpus task presented to the student. The fact that Mary was presented with a game of 15 caves, 2 bats, 2 pits and 1 stationary Wumpus was not accidental. Had she been a novice, she would have been exposed to only 1 bat, 1 pit and a warren of 10 caves. When she becomes an expert, the Wumpus is allowed to move when attacked and the number of bats and pits is increased to three.

A theory-based estimate of cognitive load is needed.

The elementary method for estimating the complexity of the game described above was added to the the coach when it became clear that the advanced game was too complex for beginners. But this method was not based on a deep theory of cognition. Our current research addresses this issue by constructing a more formal model of the *problem solver*. Our plan is to include an explicit representation of the resources required by the problem solving interpreter to apply a given set of skills. This would include such process-oriented parameters as: depth and breadth of the search space; the complexity of the data structures being maintained in terms of their size; the complexity of the patterns of individual rules in terms of the number of variables they access and and the number of conjuncts or disjuncts in their pattern; and the number of rules matching particular patterns. It would also include a representation for memory load as reflected by the number and size of the data structures that must be maintained. The extension then is to represent various limits on these resources.

There is no *a priori* reason why the load points of the Expert module's problem solving program should correspond to the load points encountered by a human problem solver. Perhaps they solve problems in very different ways. This is theoretically possible. However, remember that the problem solving skills have been formulated in a very anthropomorphic fashion. They have been

carefully broken into small pieces, comprehensible to the learner. This does not guarantee a correspondence in workload, but it suggests one. Hence, our starting point for a psychological model of the student's resource limitations will be the load points of our Expert program. There is a second rationale for this starting point. Recall that our orientation is teacher-centered. Hence, these load points are those suggested to a teacher beginning his course from a particular pespective on the syllabus. This does not obviate the need to investigate the psychological reality of this resource model: it only provides a starting point. Hence, a future goal will be to correlate this theoretically motivated formulation with the psychological literature on cognitive load. (See Norman & Bobrow, 1975 for a discussion of data-limited and resource-limited processes from a psychological perspective.)

As usual, extending the underlying problem solving representation allows the Overlay Model of the student to be more accurate. The representation for this augmented model, however, goes beyond the individual rules. An explicit model of the problem solver is required. Again we propose to employ an attribute value description. The attributes are the load dimensions: the values are the thresholds at which the student is expected to falter.

DESCRIPTION OF MARY'S PROBLEM SOLVING CAPACITY

REPRESENTATION	ESTIMATED CONFUSION THRESHOLD
NULL	SET EXCEEDS 3 ELEMENTS.
LIST	SET EXCEEDS 5 ELEMENTS.
MAP	SET EXCEEDS 15 ELEMENTS.

These thresholds are estimates based on experience with various students. The coach would have a table of such estimates for students of various age and skill backgrounds.

An interesting consequence of representing cognitive load factors is that we have a rationale for the apparently redundant data structures provided by Display Wumpus. Examine figure 6 again and observe that the caves which risk bats are represented explicitly as a list under the label BAT+ as well as by means of BAT+ symbols in the appropriate caves. Logically these representations are equivalent. But cognitively, the list-representation makes it cheap to compute how many caves are among the candidates for the bat's nest, while the map representation makes this expensive. The first requires a single data fetch on the BAT+ list while the second requires a data fetch on every cave to see if it contains the BAT+ symbol. On the other hand, the map oriented representation makes it cheap to decide if a particular cave risks bats. Thus there is a rationale for both. This is supported by the our experience that expert players maintain both kinds of data structures.

Extensions to the underlying problem solving representation also improve the utility of the Student Simulator. The performance of a particular rule model and associated data representation can be examined under different load thresholds. We plan to allow the user to set such thresholds as the maximum number of hypotheses that can be remembered implicitly with no error or the maximum number of predicates in a rule condition that can be executed with no error.

There are more steps to be taken in constructing an adequate problem solving model.

This paper has recounted the development of our problem solving model from an unorganized set of rules to a network represention of an evolving set of rules acting on an evolving data representation and executed by a resource limited problem solver. The forcing function has been the design of an adequate coaching system. Our response has been to provide the coach with multiple perspectives on its subject matter by incorporating the viewpoints of a *mathematician*, an *historian*, an *epistemologist* and a *psychologist*. The result is a teaching artifact that exhibits a deeper understanding of its subject matter as exhibited by the range of explanations it can generate.

The four steps we have described, however, do not exhaust the knowledge that a problem solving theory must represent. In this section, I enumerate several extensions, with proposals for (1) how to incorporate them into our problem solving model, and (2) how to extend the Wumpus game environment to improve its utility as an experimental base for examining these issues. These extensions are intended to incorporate within the coach the additional perspectives of the *manager*, the *learner*, the *scholar* and the *bookkeeper*.

The manager. Our discussion of problem solving ignored the *organizational skills* needed to manage large numbers of individual skills. This management includes such functions as selecting appropriate skill sets to apply to the current problem, organizing their order of application, and removing inappropriate skill sets. In the artificial intelligence literature, this class of problem solving knowledge has been explored by Davis [1978] in the context of improving the problem solving behavior of the MYCIN medical diagnosis program. Davis supplies meta-rules to represent this class of knowledge. The Wumpus rule network should be extended to incorporate this knowledge explicitly, providing, for example, an explicit representation of meta-rules for governing the order of application of individual strategies. By supplying these meta-rules, the coach could maintain an improved overlay model by measuring the use and appropriateness of various meta-

strategies in the student's play.

The learner. Our discussion of problem solving also skirted the representation of *learning skills.* A preliminary step was taken by specifying potential evolutionary relationships between rules in the skill network, but it did not characterize the learning processses involved. Again we plan to draw from the artificial intelligence literature. Newell and Moore [1973] describe the formal structure of analogies in terms of mappings between the attributes of the objects being compared. Hence, a natural extension of our rule network is to replace the labels on analogy links with descriptions of the mappings between the attributes of the connected rules. Such descriptions would record, for example, that bats and pits are analogous with respect to the distance their warnings propagate but not analogous with respect to the degree of danger which they imply. Doing so would allow the Tutor module to be more explicit in its advice about potential analogies.

The scholar. Another extension is required to represent *declarative knowledge.* We have taken a procedural viewpoint throughout this paper. Adding organizational and learning skills continues in this vein. But clearly not all of an individual's problem solving knowledge is rule-like. Such an emphasis fails to take account of the factual knowledge that an individual uses to justify the application of particular rules or to deduce those rules in the first place. The artificial intelligence literature has explored the interplay between declarative knowledge (often expressed in the predicate calculus) and procedural knowledge. For example, Green [1969] explored the derivation of programs from proofs. This class of knowledge must ultimately be included if the coach is to understand how to offer advice that emphasizes the governing principle rather than the specific rules. As a first step, we plan to include *fact nodes* in the syllabus network to represent the logical axioms of Wumpus. For example, the axiom of Wumpus that "*A cave either contains a danger or is safe, but not both.*" would be represented explicitly. This fact justifies several rules, but is not itself explicitly procedural. The fact node would be linked to the rules it justifies. By adding these fact nodes, we will improve both the overlay modelling capability of the coach and the range of advice that can be offered.

The bookkeeper. In our emphasis on rules, we have also largely ignored the *episodic structure of memory.* The coach is based on the presumption that tutoring by example is fundamental to learning. But the coach has no representation for the interrelationships between particular tutorial episodes and the rules they explain. Consistent with the four steps taken in this paper, our plan is to broaden the attribute description of individual rules by providing links to nodes describing tutorial interactions with the student. This should have a visible return by giving the coach the ability to estimate whether or not a particular rule will be remembered. The coach could base this hypothesis on the number of explanations, their frequency and their recency, all of which would be

recorded in the extended rule description.

To increase our leverage to explore these extensions, we intend to generalize the Wumpus game by multiplying its cast of characters. We plan to extend its fairy tale motif by adding such characters as dwarves, dragons, princes and princesses. Each character will have it own kind of behavior and generate its own kind of evidence. We expect that these extensions will bring into clearer focus the *organizational skills* for managing larger sets of problem solving heuristics; the *learning skills* for taking advantage of the larger number of possible analogies, generalizations and refinements that the extended world suggests; and the *memory skills* for properly organizing a more diverse set of experiences. However, these extensions still preserve the closed and tractable properties of the game environment which make it a desirable experimental domain.

A computational theory of problem solving has many educational applications.

Predicting drastic reductions in the cost of computers is now commonplace. Less clearly foreseen is their potential to perform as **problem solving tools**. Of course, a calculator is such a tool. But I have in mind a more extensive role for these machines in which they truly know something about the task and contribute accordingly. In this paper, we considered only the coaching role. However, here are three related roles that, like the coach, are based upon a computational representation for problem solving skills.

(1) Computers could serve as **personal assistants** in which the computer assumes some part of the problem solving task, thereby freeing the student to solve more complex problems. *Display Wumpus* is a simple example of such an assistant. Thorson [1978] demonstrates the fashion in which Display Wumpus can free students to reason about the logical complexity of the game without being confused by its geometric structure.

(2) Computers could provide **cognitive programming environments** in which students implement their own problem solving programs. In this fashion, students can gain a more intimate understanding of the subject matter in an active and exciting fashion. To explore this role, we are developing a *Programmable Wumpus* in which the student does not play, but rather specifies the rules to be employed by a computer player who represents him on the playing field. Thus the student acquires experience with problem solving by acting as the teacher rather than the student.

(3) Computers could provide **cognitive simulation environments** in which the consequences of various learning and teaching strategies are explored. Our *Student Simulator* is a forerunner

of this application. Potentially such simulations could serve the same role in education as *wind tunnels* do in aeronautics, namely a low cost, low risk environment for examining the behavior of scale models of students.

The potential impact of computers as problem solving tools is interesting to project, but clearly this projection must not blind us to the many difficult problems that must first be solved. These problems do not lie in building powerful hardware, but rather in developing an adequate understanding of the problem solving process. This paper has described one methodology for acquiring this understanding -- the development of computer coaches. Future research will undoubtedly include the development of consultants, assistants, and simulators as additional instruments for stressing and testing our theories of cognitive skills.

References

Brown, J. S., Burton, R. & Bell, A. SOPHIE: A step toward creating a reactive learning environment. *International Journal of Man-Machine Studies*, Vol. 7, 1975, 675-696.

Brown, J. S. & Burton, R. Diagnostic models for procedural bugs in basic mathematical skills. *Cognitive Science*, Vol. 2, 1978, 155-192.

Burton, R. & Brown, J. S. An investigation of computer coaching for informal learning environments. *International Journal of Man-Machine Studies*. 1978, in press.

Carbonell, J. AI in CAI: An artificial-intelligence approach to computer-assisted instruction. *IEEE Transactions on Man-Machine Systems*, Vol. MMS-11, No. 4, December 1970.

Carr, B. Wusor II: A computer aided instruction program with student modelling capabilities. *MIT Artificial Intelligence Laboratory Memo 417*, May 1977.

Carr, B. & Goldstein, I. Overlays: A theory of modelling for computer aided instruction, *MIT Artificial Intelligence Laboratory Memo 406*, February 1977.

Clancey, B. Tutoring rules for guiding a case method dialogue, *International Journal of Man-Machine Studies*, 1978, in press.

Davis, R. Applications of meta level knowledge to the construction, maintenance and use of large knowledge bases. *Stanford AI Memo 283*, July 1976.

Goldstein, I. & Carr, B. The computer as coach: an athletic paradigm for intellectual education. *Proceedings of 1977 Annual Conference, Association for Computing Machinery*, Seattle, October 1977, 227-233.

Goldstein, I.P. The genetic epistemology of rule systems. *International Journal of Man-Machine Studies*. 1978, in press.

Green, C. Application of theorem proving to problem solving. *Proceedings of the first International Joint Conference on Artificial Intelligence*, 1969, 219-239.

Miller, M. A structured planning and debugging environment for elementary programming. *International Journal of Man-Machine Studies*, in press.

Moore, J. & Newell, A. How can Merlin understand. In L. Gregg (Ed.), *Knowledge and Cognition.* Potomac, Md.: Lawrence, Erlbaum Associates, 1973.

Newell, A. & Simon, H. *Human Problem Solving*, New Jersey: Prentice Hall, 1972.

Norman, D. A. & Bobrow, D. G. On data-limited and resource-limited processes. *Cognitive Psychology, Vol. 7, 44-64.*

Pólya, G. *How to Solve It.* New York: Doubleday Anchor Books, 1957.

Sleeman, D. A problem-solving monitor for a deductive reasoning task. *International Journal of Man-Machine Studies*, Vol. 7, 1975 183-211.

Stansfield, J., Carr, B., & Goldstein, I. Wumpus Advisor I: a first implementation of a program that tutors logical and probabilistic reasoning skills. *MIT AI Laboratory Memo 381*, Sept. 1976.

Thorson, Linda. A comparative study of the [illegible]fect of a display on the performance of players of a computer game. *MIT AI Working P[illegible]* in preparation.

Yob, G. Hunt the Wumpus. *Creative C[illegible]pu[illegible]g*, September/October, 1975, 51-54.

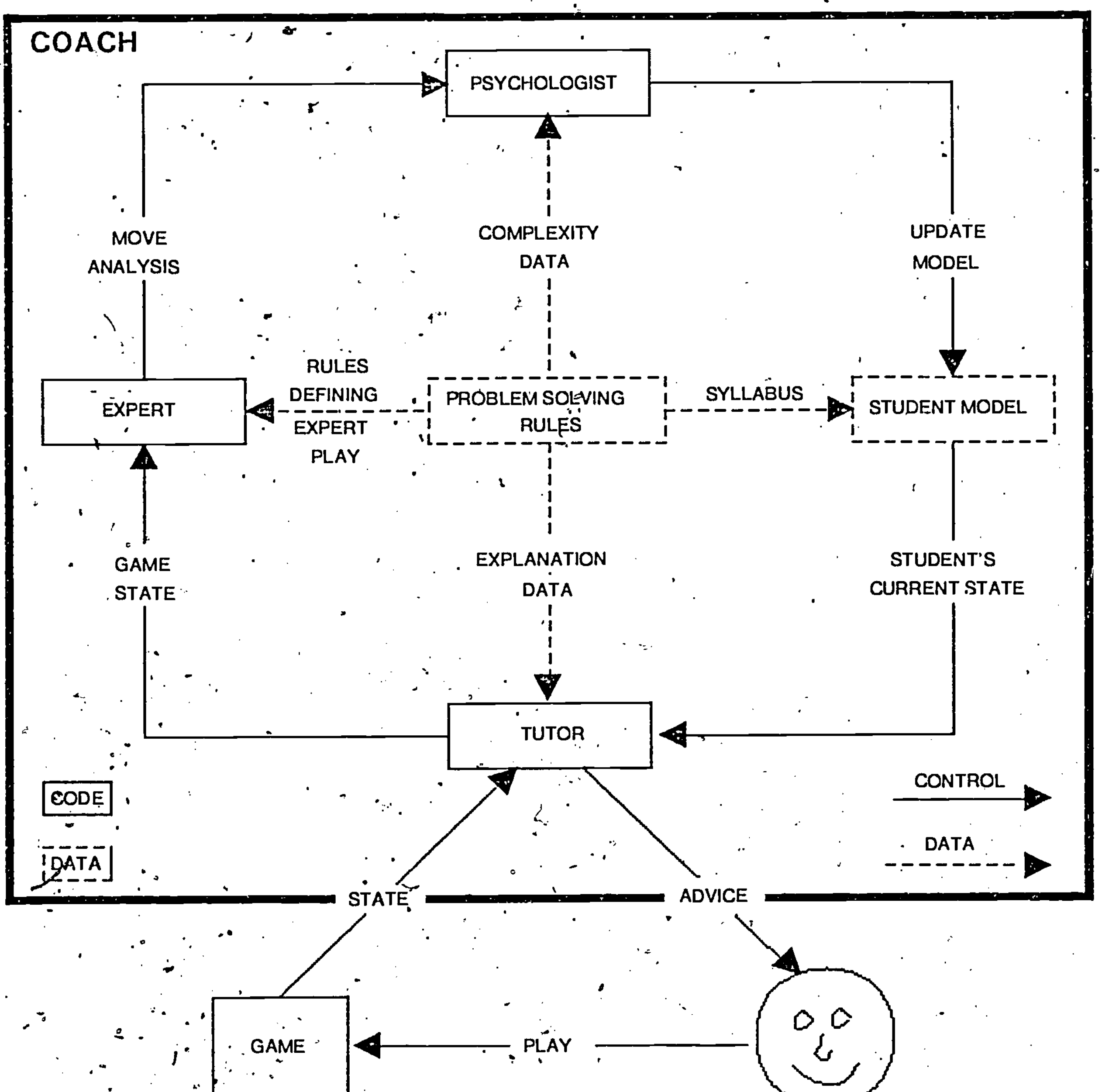


Fig. 1. A simpified, block diagram of a computer coach.

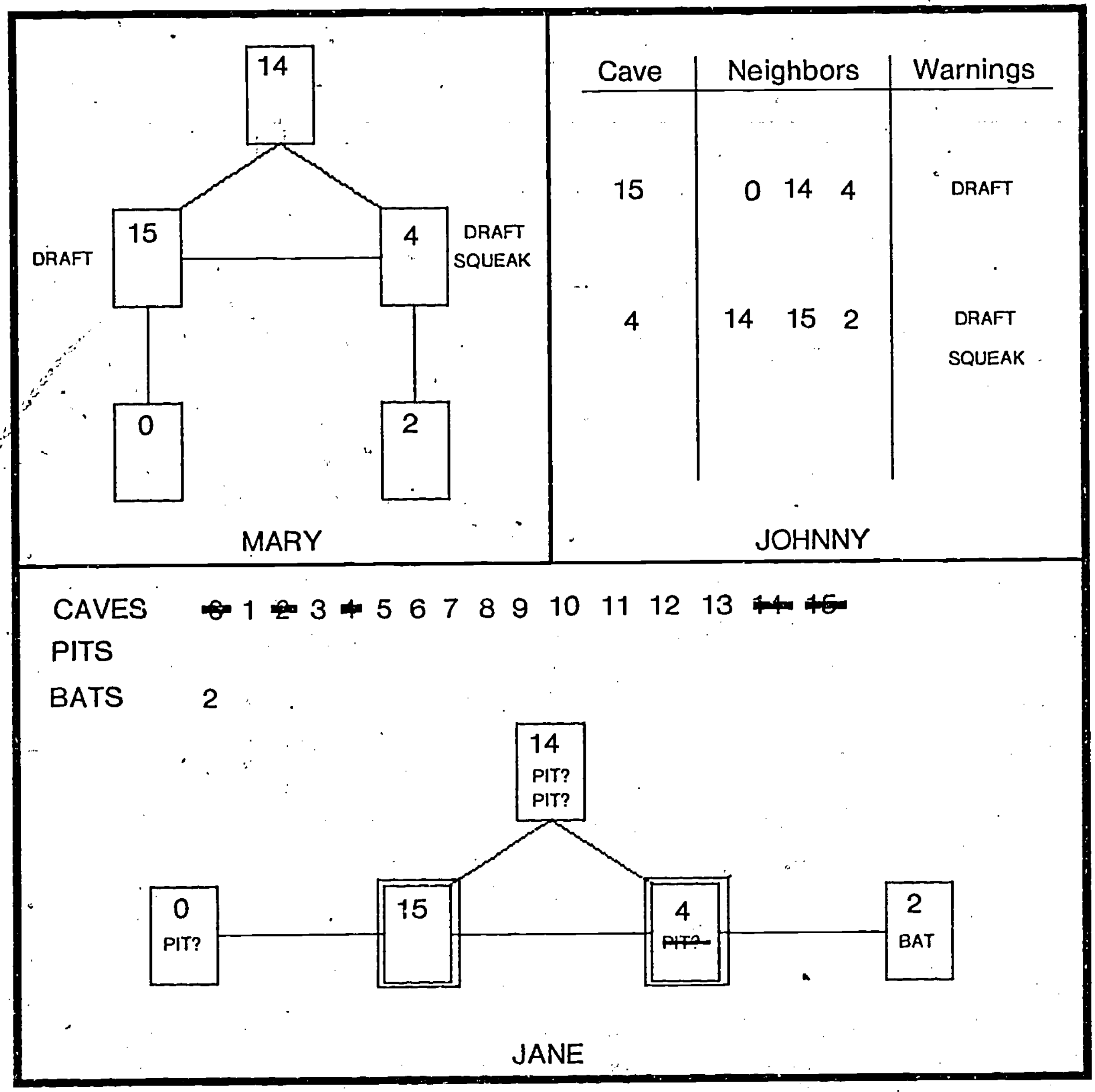
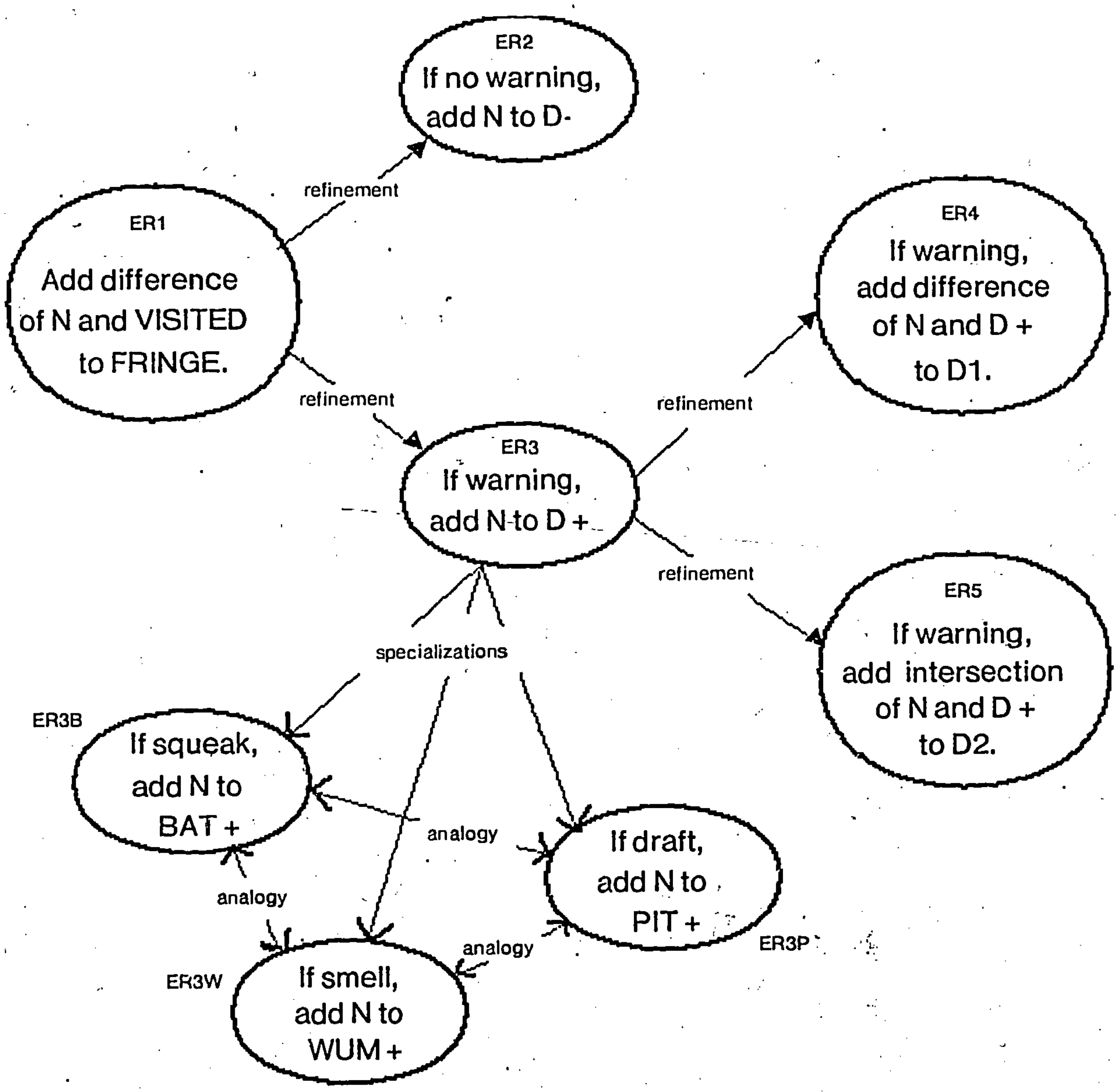


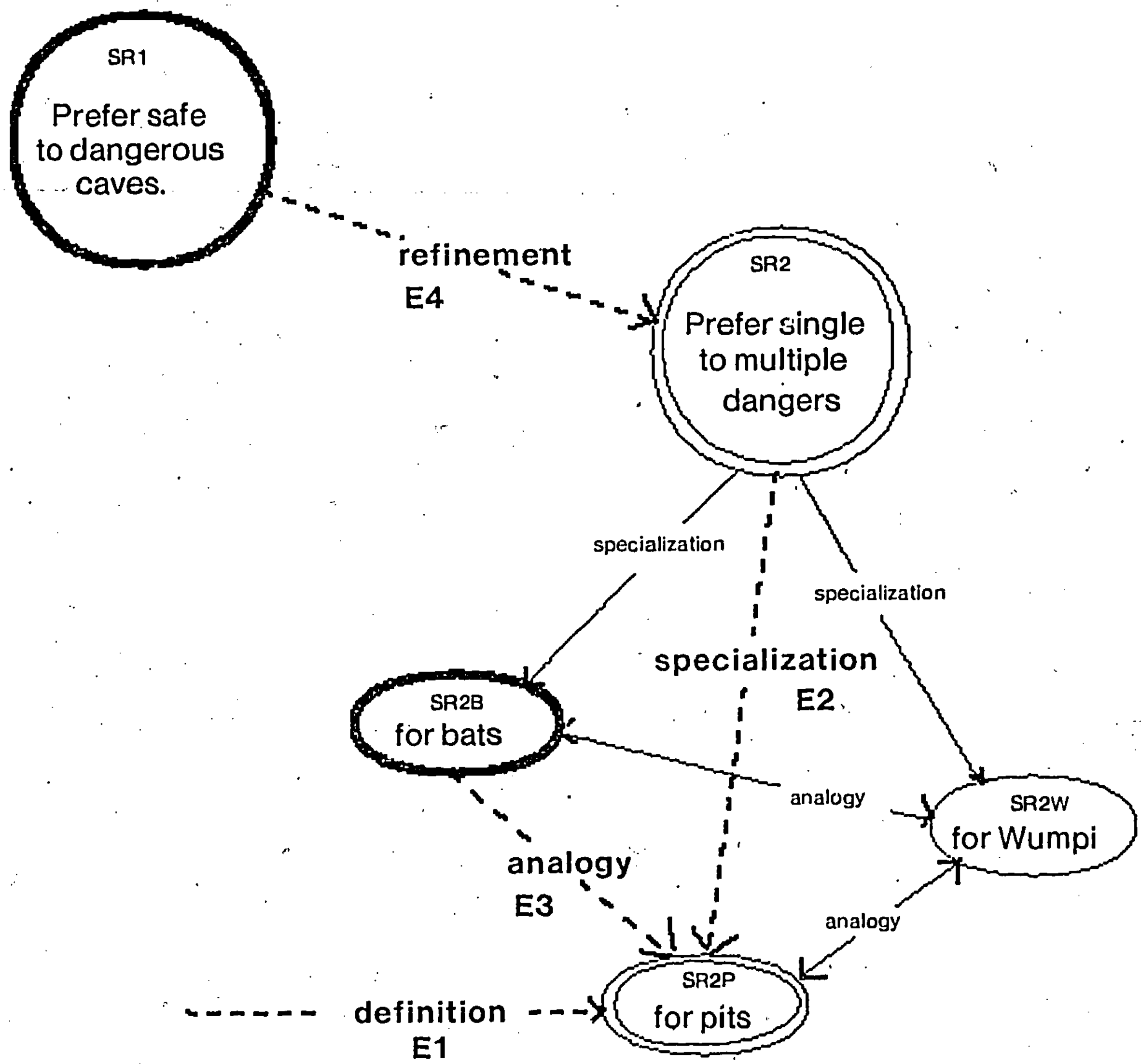
Fig. 2. Player's representations of the game.



N is the set of neighbors of the current cave.
VISITED is the set of caves that have been visited.
FRINGE is the set of unvisited caves adjacent to visited caves.

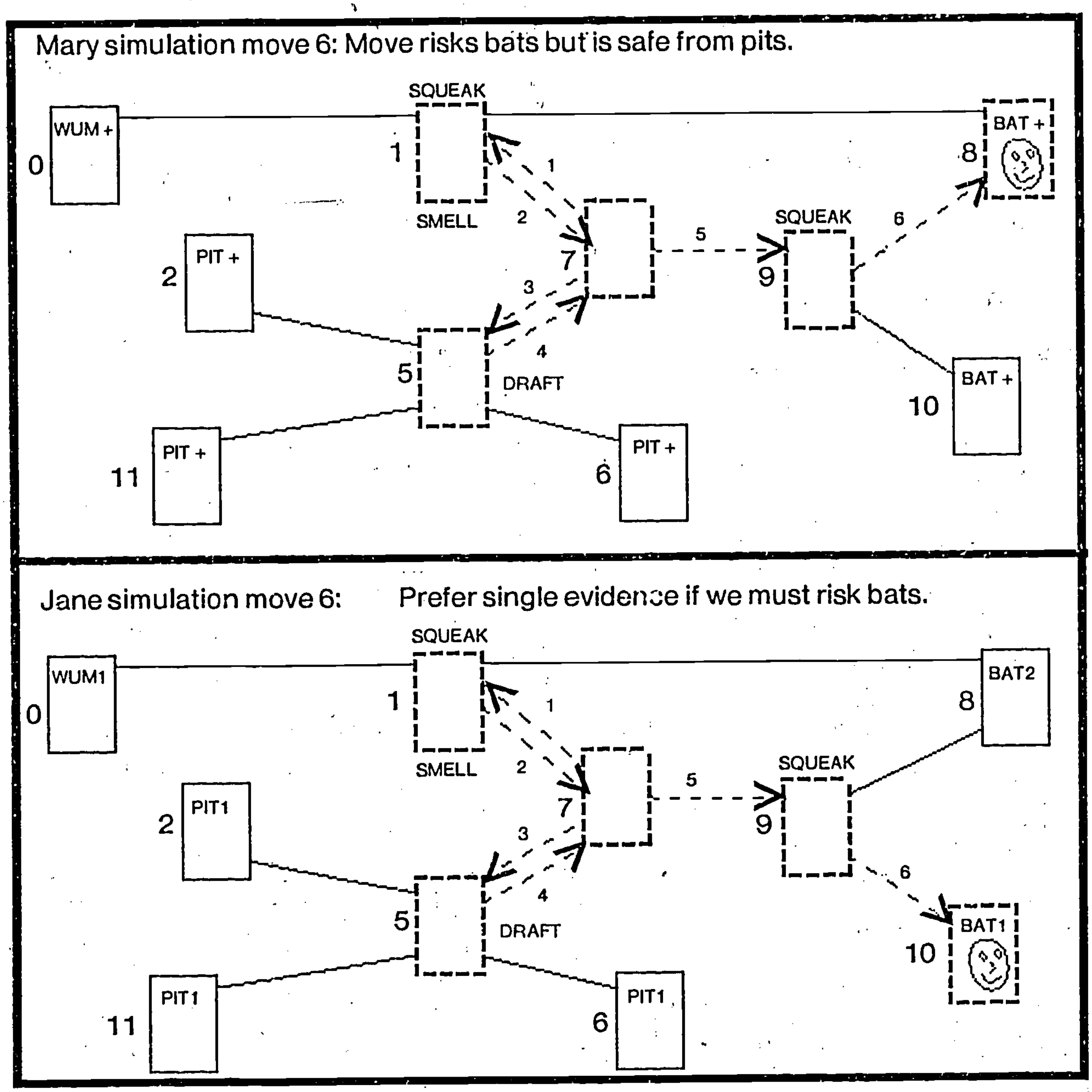
D + are sets of caves that possibly risk a danger.
D- are sets of caves that are safe from a danger.
D1 are sets of caves for which there is single evidence of a danger.
D2 are sets of caves for which there is double evidence of a danger.

Fig. 3. A region of the Wumpus skill network concerned with evidence rules.



A dark outline signifies a rule hypothesized to be known by the player.
A double outline signifies a rule explained to the player.
A dashed link signifies that an explanation corresponding to that link has been generated.
E1 through E4 are the corresponding explanations in the text.

Fig. 4. An overlay model for a region of the skill network.



Dashed arrows signify the path of the simulated student.

Fig. 5. Divergent behavior of two simulated students.

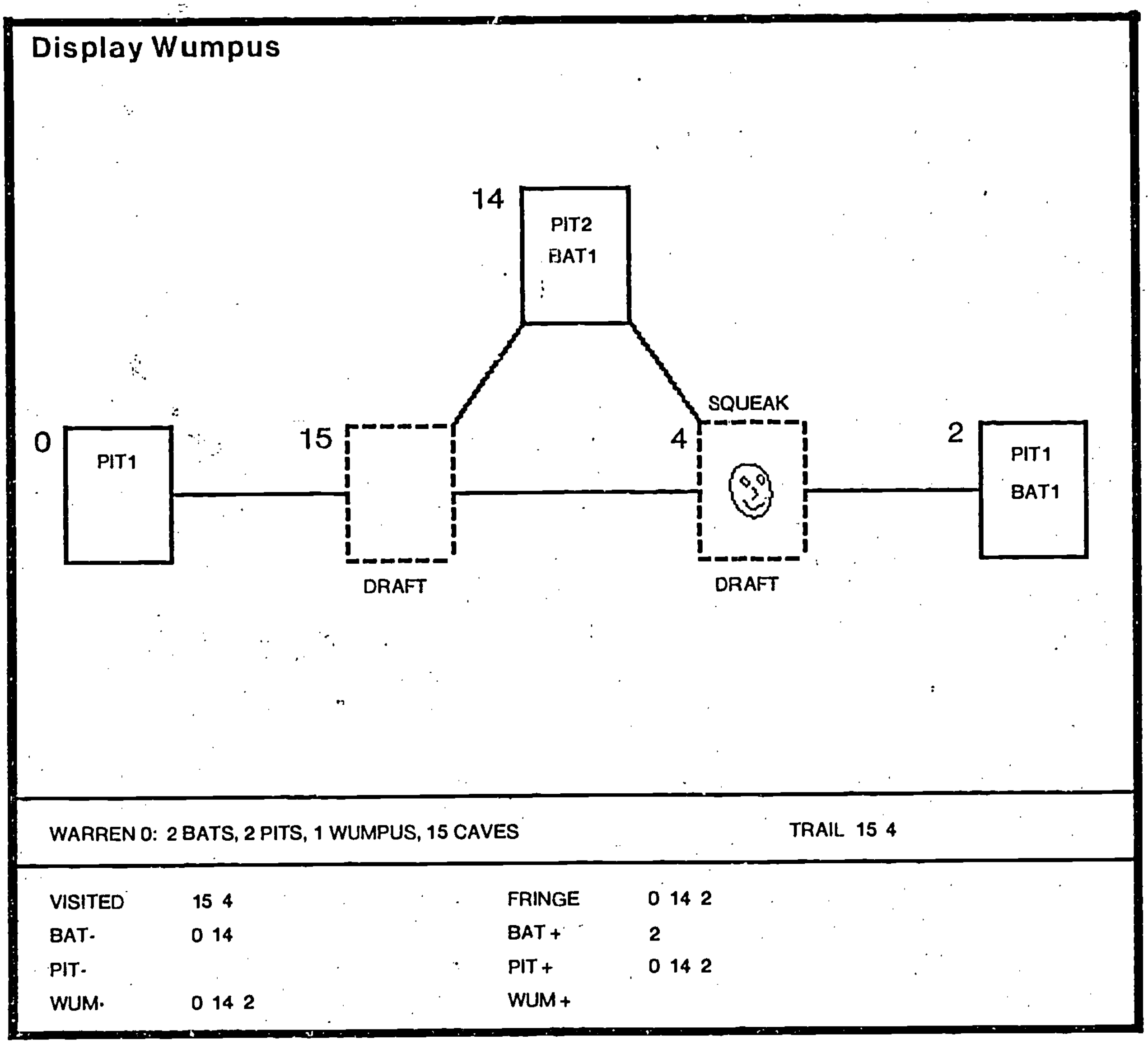


Fig. 6. The map and list representations provided by Display Wumpus.

SEDR DATA FORMAT

1. Award Number	2. Award Date	3. Termination Date
SFD77-19279	September 23, 1977	February 29, 1980

6. Title
A Research Study of Comput›r-Based Tutoring of Mathematical and Scientific Knowledge

4. Amount of Award
$193,400 (Cum. Amt.)

5. Type
Final Tech. Report

7. Performing Organization
Massachusetts Institute of Technology
77 Massachusetts Avenue
Cambridge, MA 02138

8. Pagination
31 pages

9. Accession Number
00231

11. Principal Investigator; Field or Specialty
Ira P. Goldstein [Electrical Enginering]

10. Performing Organization Report Number

12. NSF Program Manager
Andrew P. Molnar

13. SEDR Program
RISE

14. SEDR Subprogram

15. Abstract

A formal model of the syllabus and an associated set of tutorial strategies were developed as part of an investigation of the teaching of mathematical skills. A computer coach for an elementary probability game was employed to test the theory. Computer coaching is a type of computer assisted instruction in which the coaching program observes the performance of a student engaged in some intellectual game. The coach's function is to intervene occasionally in student generated situations to discuss appropriate skills that might improve the student's play. Coaching is a natural context in which to investigate the teaching and learning processes, but it is a demanding task. The computer must be able to analyze the student's performance in terms of a model of the underlying problem solving skills. This model must represent not only expertise for the task but also intermediate stages of problem solving skill and typical difficulties encountered by the learner. Implementing several generations of computer coaches to meet these demands has resulted in a model that represents problem solving skills as an evolving set of rules for a domain acting on an evolving representation of the problem and executed by a resource-limited problem solver.

16. Descriptors
Computer Assisted Instruction
Information Processing
Artificial Intelligence
Cognitive Processes

17. Identifiers
Knowledge Representation
Computer Games, Wumpers

18. Field
Software Systems / Computer Science

19. Target Audience
Grades 7-12

20. Availability

21. Supplemental Notes

NATIONAL SCIENCE FOUNDATION
Washington, D.C. 20550

FINAL PROJECT REPORT
NSF FORM 98A

6-23-80

PLEASE READ INSTRUCTIONS ON REVERSE BEFORE COMPLETING

PART I—PROJECT IDENTIFICATION INFORMATION

1. Institution and Address	2. NSF Program	3. NSF Award Number
Massachusetts Institute of Technology 77 Mass. Ave. Cambridge, MA 02138	Rise 4. Award Period From 9/1/77 To 8/31/79	SED77-19279 5. Cumulative Award Amount 193446

6. Project Title
A research study of computer-based tutoring of mathematical and scientific knowledge.

PART II—SUMMARY OF COMPLETED PROJECT *(FOR PUBLIC USE)*

A formal model of the syllabus and an associated set of tutorial strategies were developed as part of an investigation of the teaching of mathematical skills. A computer coach for an elementary probability game was employed to test the theory. Computer coaching is a type of computer assisted instruction in which the coaching program observes the performance of a student engaged in some intellectual game. The coach's function is to intervene occasionally in student generated situations to discuss appropriate skills that might improve the student's play. Coaching is a natural context in which to investigate the teaching and learning processes, but it is a demanding task. The computer must be able to analyze the student's performance in terms of a model of the underlying problem solving skills. This model must represent not only expertise for the task but also intermediate stages of problem solving skill and typical difficulties encountered by the learner. Implementing several generations of computer coaches to meet these demands has resulted in a model that represents problem solving skills as an evolving set of rules for a domain acting on an evolving representation of the problem and executed by a resource-limited problem solver.

PART III—TECHNICAL INFORMATION *(FOR PROGRAM MANAGEMENT USES)*

1. ITEM *(Check appropriate blocks)*	NONE	ATTACHED	PREVIOUSLY FURNISHED	TO BE FURNISHED SEPARATELY TO PROGRAM: Check (✓)	TO BE FURNISHED SEPARATELY TO PROGRAM: Approx. Date
a. Abstracts of Theses	X				
b. Publication Citations		X			
c. Data on Scientific Collaborators		X			
d. Information on Inventions	X				
e. Technical Description of Project and Results		X			
f. Other *(specify)*					

2. Principal Investigator/Project Director Name *(Typed)*	3. Principal Investigator/Project Director Signature	4. Date
Ira P. Goldstein	Ira P. Goldstein	5/8/80

NSF Form 98A (5-78) Supersedes All Previous Editions

Form Approved OMB No. 99R0013

Collaborators

Mark Jeffery, graduate student

Mark Miller, graduate student

Sandor Schoichett, graduate student

James Stansfield, research staff

Linda Thorson, research staff

Barbara White, graduate student

Publications

Journal Articles

Goldstein, I.P., "The Genetic Epistemology of Rule Systems", *International Journal for Man Machine Studies*, vol. 11, no. 1., January, 1979, pp. 51-78. Also MIT AI Memo 449, January 1978.

Book Articles

Goldstein, I. P., "Developing a Computational Representation for Problem Solving Skills" in Reif, F. and D. T. Tuma (eds.), *Problem Solving and Education: Issues in Teaching and Research*, Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1980, pp. 53-80.

Miller, M. and I.P. Goldstein, "Planning and Debugging in a Computer Coach for Elementary Programming", in P. H. Winston (ed.), *Artificial Intelligence: an MIT Perspective*, MIT Press, Cambridge, 1979, 317-37.

Goldstein, I.P. and J.S. Brown, "The Computer as a Personal Assistant for Learning" in J. Lockhead and J. Clement (eds.), *Cognitive Process Instruction*, Franklin Institute Press, Philadelphia, 1979, 201-207.

Conference Articles

Goldstein, I. P., "The Implications of a Formal Model of the Syllabus for a Theory of Learning," *Proceedings of the National Conference of the Association for the Development of Computer Instructional Systems*, March 1979, San Diego.

Goldstein, I.P., "The Genetic Graph: A Representation for the Evolution of Procedural Knowledge", *Proceedings of the Second National Conference of the Canadian Society for Computational Studies of Intelligence*, July 1978.

Brown, J. S. and I.P. Goldstein, "Intelligent Instructional Systems," *Congressional Record*, House Science and Technology Subcommittee on Domestic and International Scientific Planning, Analysis, and Cooperation. October 12, 1977.