DOCUMENT RESUME

ED 151 013 IR 005 728

AUTHOR	Burton, Richard R.; Brown, John Seely
TITLE	A Tutoring and Student Modelling Paradigm for Gaming Environments.
INSTITUTION	Bolt, Beranek and Newman, Inc., Cambridge, Mass.
PUB DATE	Feb 76
NOTE	13p.; Best copy available
EDRS PRICE	MF-$0.83 HC-$1.67 Plus Postage.
DESCRIPTORS	*Computer Assisted Instruction; *Educational Games; *Elementary Education; Feedback; Man Machine Systems; *Mathematics Instruction; Models; *Programed Tutoring; Simulation; Systems Analysis

ABSTRACT

This paper describes a paradigm for tutorial systems capable of automatically providing feedback and hints in a game environment. The paradigm is illustrated by a tutoring system for the PLATO game "How the West Was Won." The system uses a computer-based "Expert" player to evaluate a student's moves and construct a "differential model" of the student's behavior with respect to the Expert's. The essential aspects of the student's behavior are analyzed with respect to a set of "issues," which are addressed to the basic conceptual constraints that might prevent the student's full utilization of the environment. Issues are viewed as procedural specialists that "wake-up" or become active when an instance of an issue manifests itself in a move. These issue specialists help the Tutor isolate what to comment on. The intent of the system is to transform a "fun" game into a productive learning environment without altering the student's enjoyment. (Author)

A TUTORING AND STUDENT MODELLING PARADIGM FOR GAMING ENVIRONMENTS

by

Richard R. Burton
and
John Seely Brown

A TUTORING AND STUDENT MODELLING PARADIGM FOR GAMING ENVIRONMENTS

Richard R. Burton and John Seely Brown
Bolt Beranek and Newman Inc.
Cambridge, Massachusetts

Abstract

This paper describes a paradigm for tutorial systems capable of automatically providing feedback and hints in a game environment. The paradigm is illustrated by a tutoring system for the PLATO game "How the West Was Won". The system uses a computer-based "Expert" player to evaluate a student's moves and construct a "differential model" of the student's behavior with respect to the Expert's. The essential aspects of the student's behavior are analyzed with respect to a set of "issues", which are addressed to the basic conceptual constraints that might prevent the student's full utilization of the environment. Issues are viewed as procedural specialists that "wake-up" or become active when an instance of an issue manifests itself in a move. These issue specialists help the Tutor isolate what to comment on. The intent of the system is to transform a "fun" game into a productive learning environment without altering the student's enjoyment.

TABLE OF CONTENTS

I. INTRODUCTION

An exciting and relatively unexplored use of computers in education involves coupling an adaptive tutor (or commentator) to an educational game. Games provide an enticing problem solving environment which the student can explore at will, creating his own ideas of its underlying structure and synthesizing strategies which reflect his understanding of this structure. Games also have the potential for motivating drill and practice by providing environments in which students actually enjoy repetition. However, in both cases, a major stumbling block to the effective use of such "unstructured gaming environments" is the amount of teacher attention that is often required to keep the student from forming grossly incorrect models of the underlying structure of the game and to identify interesting shortcomings of particular strategies. In brief, for a gaming environment to be fully utilized as a learning instrument the environment must be augmented by tutorial guidance which points out weaknesses in the student's ideas or suggests ideas when the student appears to have none. This paper presents a paradigm for designing computer systems capable of providing this kind of tutorial guidance and describes an example of one such system built around a drill and practice game in arithmetic.

Before describing our basic paradigm for constructing tutoring systems, we want to stress the difference between the notion of tutorial behavior as used here and that which has previously been proposed. In classical CAI, the tutoring behavior is locally controlled by predetermined branching points in an instructional sequence. The instructional sequence is restricted to the extent that each branching point is testing for the understanding of a small number of concepts. The author of the sequence is then able to predict which misconceptions lead to what responses and branch to the proper remedial sequences. In the gaming environment, the course of the game is determined largely by the student. The tutoring module is given freedom to interrupt the student at any time and make suggestions or correct misconceptions, but it cannot take control of the game away

To appear in "Proceedings for the Symposium on Computer Science and Education" Anaheim, Calif., Feb. 1976.

from the student. That is, the tutoring module has no specified branching points or any other explicit (as opposed to implicit) means for directing the game into particular situations. Hence a major challenge in creating this kind of tutor is to enable it to use its knowledge of the domain together with a synthesized model of the student's past behavior to decide what to say and when to say it. The tutor must be perceptive enough to make relevant comments but at the same time it must not be so intrusive as to destroy the fun inherent in the game.

The viability of this approach depends critically on techniques for automatically inducing a "model" of the student which accurately represents his reasoning strategies and current state of knowledge. If the computer-based tutor is to deviate from (or not to use at all) a predetermined instructional sequence, its new course of action must be based not only on its reasoning capabilities but also on the details of a student's observed strengths and weaknesses and any shortcomings manifested in his current "move."

Tutoring by Issue and Example -- a General Paradigm

The paradigm of "issues and examples" was developed to focus the tutoring system on relevant portions of the student's behavior. The important aspects (skills or concepts) of the domain (i.e., what the student is expected to know or learn) are identified as a collection of "issues". The issues determine what parts of the student's behavior are monitored by the tutor. Each issue is activated by patterns which watch the student's behavior for evidence that the student uses or does not use their particular concept or skill. As the student plays, a model of how he is performing, with respect to each issue, is constructed. When he makes a "bad" move, a tutorial program uses the model to decide why the student did not make a better move, that is, which issue he missed. Once an issue has been determined, the tutor can present an explanation of that issue together with a better move which illustrates the issue. In this way, the student can see the usefulness of the "issue" at a time when he will be most receptive to the idea presented -- immediately after he has thought about the problem.

Figure 1 is a diagram of the modelling/tutorial process underlying the paradigm. Figure 1a presents the process of constructing of 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, he exhibits a certain behavior. The important aspects of this behavior (the issues) are abstracted by the pattern matching component of each issue (called the "recognizer"). This abstracting is also done with respect to 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. Notice that without the Expert it is not possible to determine whether the student is weak in some area or whether the need for that skill has arisen infrequently in the student's experience.

Figure 1. Diagram of the Modelling/Tutorial Process

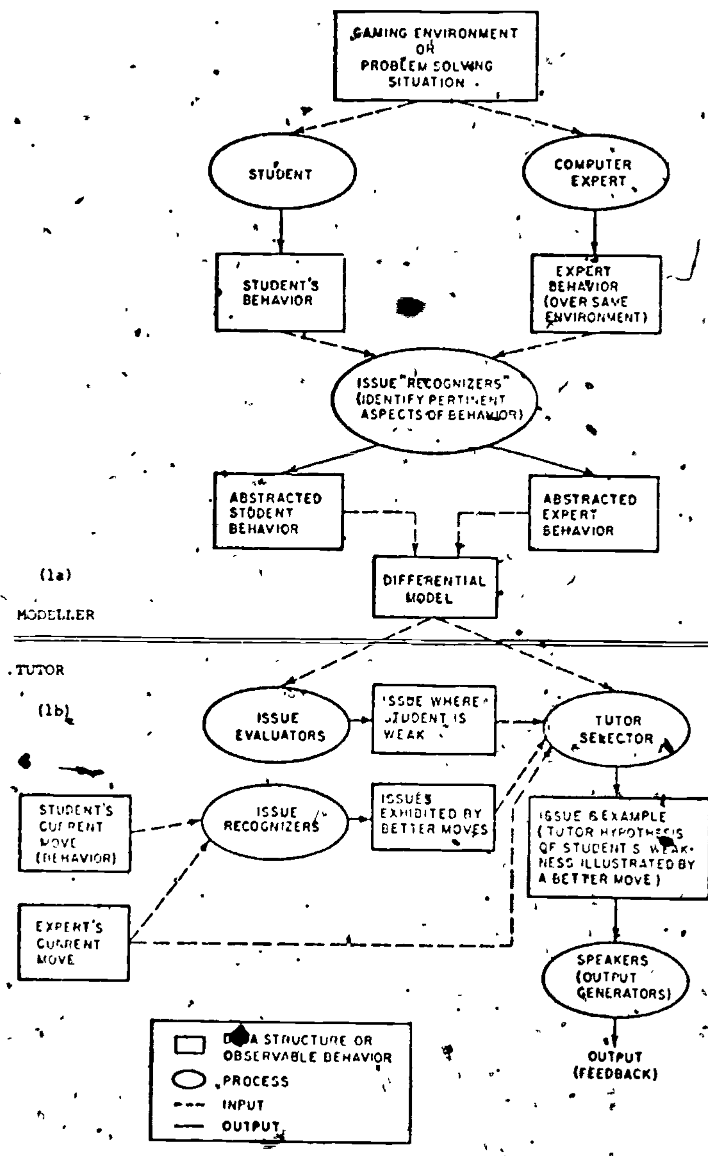


Figure 1b presents the tutoring process. When the student makes a less than optimal move (as determined by comparing his move with those of the Expert), the Tutor uses the model evaluation component of each issue (called the "evaluator") to scan the student model and to create a list of issues on which the student is weak. From the Expert's list of better moves, the tutor uses 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 tutor selects an issue and a good move which illustrates it. The selected issue and example are then passed to the output generators which produce the feedback to the student.

We would like to stress two points in the above process. One is the necessity of the Expert and the other is the importance of identifying the critical issues. The Expert provides a measure for evaluating the student's behavior in novel situations without which it would be necessary to severely restrict the game situations which could be tutored. The issues define those conceptual components of the environment which the student is expected to learn and they provide the tutor a handle to structure and direct the exploration of the environment by the student.

II. AN EXAMPLE SYSTEM

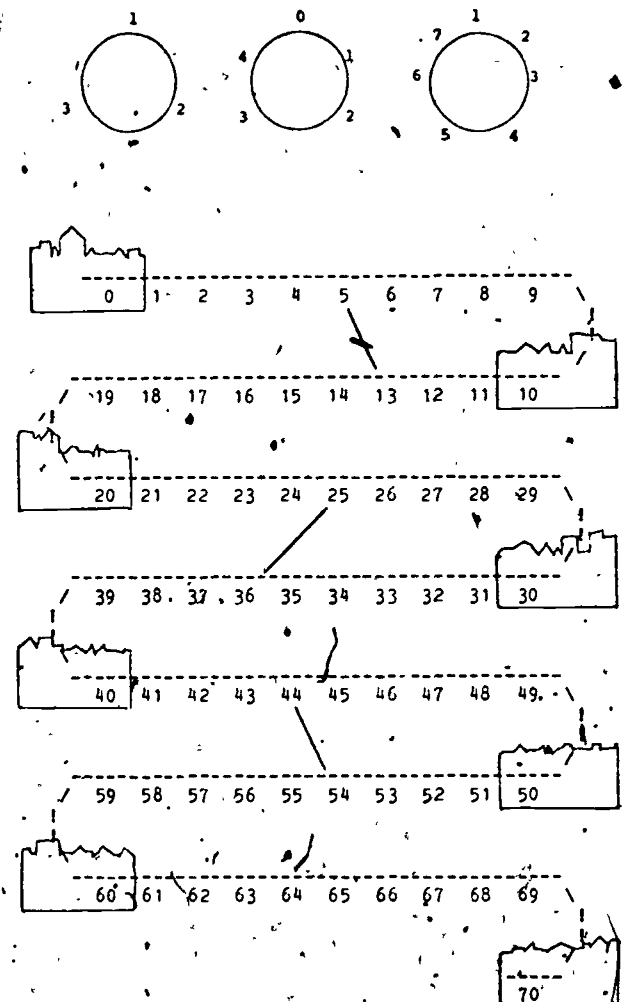
In order to explore the ramifications and effectiveness of the "issue and example" paradigm, we chose a domain in which we could easily construct an expert program that the tutor could call on for evaluating the student's behavior. The domain of knowledge chosen was the PLATO game "How the West Was Won."*

Description of "How the West Was Won"

"How the West Was Won" (hereafter called West) is a game for two players (the computer usually being one). It is played on a game board like that in Figure 2. The object of the game is to get to the last town on the map (position 70). On each turn a player gets three spinners (random numbers). He can combine the values of the spinners using any two (different) arithmetic operators (+, -, * or /). The value of the arithmetic expression he makes is the number of spaces he gets to move. (He must also say what the answer is.) If he makes a negative number, he moves backwards.

Along the way there are shortcuts and towns. If a player lands on a shortcut, he advances to the other end (e.g. from 5 to 13 in Figure 2). If he lands on a town, he goes on to the next town. When a player lands on the same place as his opponent, unless it is a town, his opponent must retreat back two towns. To win, a player must be the first one to land exactly on the last town. Both players get the same number of turns, so ties are possible.

Figure 2. Game board for How the West Was Won (from PLATO)



---

*This game was written by Bonnie Anderson for the PLATO Elementary Mathematics Project.

Why Tutor at All?

A central assumption of the West tutoring system is that good tutoring can point out structure in an environment which might have otherwise been missed and by so doing allow the student to enrich his understanding of (and skills in) the environment. In Plato West, an untutored (unwatched) student tends to become fixed on a subset of the available moves and hence misses the potential richness of the game. 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 the 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 number). If this strategy is augmented by a rule that prevents moving off the board (a simple end game strategy) it generates a respectable game. Notice, however, that much is missed. The student is unaware of the special moves such as bumps and therefore of such questions as, "Is it better to send my opponent back 14 or get 9 ahead of him?" In fact, since this kind of student strategy lets him consider only one move, he misses the whole notion of strategies for deciding between alternative moves. From an arithmetic drill and practice point of view, he is performing one calculation per move instead of dozens of mental calculations which he would have to perform to answer questions such as, "What numbers can I form with these spinners?" or "Can I make an 8 with these spinners?" By interjecting comments and suggesting better moves to the student from time to time (though not too often), the tutor tries to widen the student's view of the game, hence drastically improving the drill and practice aspect by getting him to compute many more expressions per move.*

---

*Much of our original motivation for building the West tutoring system stemmed from the doctoral thesis of Cecily Resnick [1975] which describes some preliminary experiments questioning the effectiveness of this game as a learning or drill and practice environment. Resnick found that although students become intensely involved in the game they usually developed a limited strategy and would play the game for hours using their strategy regardless of how many games they won or lost. Furthermore, the strategies they would lock onto often required no search and hence only one arithmetic expression per move was being evaluated.

Protocol

Before discussing the modelling/tutoring process in greater detail, we present a protocol of an actual student playing our West. The tutoring component of the system used for the protocol was accelerated to generate more feedback than normal. The normally paced tutor seldom hassles the student two moves in a row. In the protocol, all lines typed by the student begin with "=> ". Annotations are indented.

```
WELCOME TO "HOW THE WEST WAS WON".
WHAT IS YOUR NAME?
=> BOB
DO YOU WANT TO GO FIRST?
=> YES

IT'S YOUR TURN... SPINNERS ARE: 1 2 2
WHAT ARITHMETIC EXPRESSION DID YOU FORM?
=> 1*2+2
WHAT NUMBER DOES THAT GIVE YOU?
=> 4
```

> The numbers that could have been made were:
> -2 2 -1 -3 5 3 0 4 6 1
> "best" move is (2*2)+1.
> delta-ways: 13-1, 10-8, 6-1, 4-5, 3-2, 2-3, 1-3
> Move ranks 11 which is FAIR
>
> The indented information above is generated by the EXPERT module on each turn. This information is not seen by the student but is used by the TUTOR, MODEL EVALUATOR and STUDENT MODELLER modules. It is included here to indicate the complexity of information that is used by the system to follow the student's behavior. The first line gives the range of numbers that could legally be made using the spinners. The second line gives the "best" move according to the metric of maximizing the difference between you and your opponent. (These two pieces of information are available to the student through the "hint" key if he wants it.) The third line (beginning "delta-ways") represents the legal move space. The first number in each pair gives the number of spaces ahead the student would be (i.e. delta). The second number is the number of different arithmetic patterns which could have been used to achieve that lead. In this case there is one move which would have put Bob 13 ahead (making 5 and taking the shortcut), eight moves which would have put him 10 ahead (making 0 and getting advanced to the next town), one that would put him 6 ahead, etc. His move (4 ahead), ranked 11 because there were 10 better ones. For the rest of the protocol only the Rank of the student's move will be shown.

```
BOB IS AT 4 -- THE COMPUTER IS AT 0

IT'S MY TURN.. SPINNERS ARE: 1 2 3
MY EXPRESSION IS: (1*2)+3
WHICH GIVES ME 5
I TOOK A SHORTCUT
THE COMPUTER IS AT 13 -- BOB IS AT 4

IT'S YOUR TURN.. SPINNERS ARE: 2 1 2
WHAT ARITHMETIC EXPRESSION DID YOU FORM?
=> 2*1+2
WHAT NUMBER DOES THAT GIVE YOU?
=> 4
```

Bob makes another less-than-optimal move but two moves is too early to start tutoring.

Move ranks 9 which is FAIR

```
BOB IS AT 8 -- THE COMPUTER IS AT 13

IT'S MY TURN.. SPINNERS ARE: 1 2 1
MY EXPRESSION IS: (1+1)*2
WHICH GIVES ME 4
THE COMPUTER IS AT 17 -- BOB IS AT 8.

IT'S YOUR TURN.. SPINNERS ARE: 1 1 5
WHAT ARITHMETIC EXPRESSION DID YOU FORM?
=> 1*1+5
WHAT NUMBER DOES THAT GIVE YOU?
=> 6
```

Move ranks 3 which is GOOD
The ISSUES are: PATTERN PARENS STRATEGY

These are the tutorable issues which have been determined by considering the student model together with the moves which are better. Bob's first three moves have all had the same pattern, a*b+c which does not require parentheses; whereas the best move uses a different pattern which does require parentheses so the issues of PATTERN and PARENS are noticed. The issue of PATTERN was chosen because it is more specific than PARENS. That is, PATTERN appears ahead of PARENS on a printing list. The STRATEGY issue indicates that there is no coherent strategy arising for the series of moves made by the student, as will be discussed later.

```
IT'S BEGINNING TO SEEM LIKE YOU STICK TO
THE SAME OLD STANDARD PATTERN IN YOUR MOVE.
THE REALY EXPERT PLAYERS TRY A WHOLE BUNCH
OF POSSIBILITIES. AN EXAMPLE OF A GOOD
EXPRESSION WOULD BE: (1+1)*5, WHICH IS A
PRETTY GOOD MOVE. SO YOU COULD HAVE BEEN
AT 18 WITH ME FINISHING UP THE TURN AT 17.
```

After the presentation of a better move, the student may be allowed to retake his turn. This usually makes the student more receptive to the advice. In this case Bob decides not to put himself in front of his opponent.

```
WOULD YOU LIKE TO TAKE YOUR TURN OVER?
=> NO
BOB IS AT 14 -- THE COMPUTER IS AT 17

IT'S MY TURN.. SPINNERS ARE: 3 3 2
MY EXPRESSION IS (3+2)*3
WHICH GIVES ME 15
THE COMPUTER IS AT 32 -- BOB IS AT 14

IT'S YOUR TURN.. SPINNERS ARE: 3 4 5
WHAT ARITHMETIC EXPRESSION DID YOU FORM?
=> 3+4*5
WHAT NUMBER DOES THAT GIVE YOU?
=> 23
```

Move ranks 7 which is FAIR
The ISSUES are: PARENS ORDER STRATEGY

In this case the tutor picks on the issue of parentheses use. ORDER (using the spinner in a particular order) was not an issue last turn because none of the better moves required changing the order. (In a less condensed session, the tutor would not comment on two moves in a row.)

```
I NOTICE THAT YOU DON'T USE PARENTHESES
VERY OFTEN.. I THINK YOU COULD WIN MORE
OFTEN IF YOU TRIED USING PARENTHESES. FOR
EXAMPLE, YOU COULD HAVE MADE THE
EXPRESSION: (3+4)*5, WHICH IS ALRIGHT FOR
AN AMATEUR! SO YOU COULD HAVE BEEN AT 49
AND I WOULD ONLY HAVE BEEN AT 32.

WOULD YOU LIKE TO TAKE YOUR TURN OVER?
=> YES
```

In this case Bob retakes his turn. The players are put in their original position and the spinners are the same. Retaking a turn allows the student to go through the motions of the suggested move for himself and makes it easier to remember.

```
BOB IS AT 14 -- THE COMPUTER IS AT 32
IT'S YOUR TURN.. SPINNERS ARE: 3 4 5
WHAT ARITHMETIC EXPRESSION DID YOU FORM?
=> (3+4)*5
WHAT NUMBER DOES THAT GIVE YOU?
=> 35
BOB IS AT 49 -- THE COMPUTER IS AT 32

IT'S MY TURN.. SPINNERS ARE: 2 2 1
MY EXPRESSION IS: 2*(1-2)
WHICH GIVES ME -2
I LANDED IN A TOWN.
THE COMPUTER IS AT 40 -- BOB IS AT 49

IT'S YOUR TURN.. SPINNERS ARE: 1 2 3
WHAT ARITHMETIC EXPRESSION DID YOU FORM?
=> 2+1*3
WHAT NUMBER DOES THAT GIVE YOU?
=> 5
```

Move ranks 13 which is POOR
The ISSUES are: DIVIDE MINUS STRATEGY

Up to this point the only operations Bob has used are addition and multiplication, so division and subtraction become issues because there are better moves which use them.

YOU SHOULDN'T FORGET THAT IT'S ALRIGHT TO DIVIDE, SO LONG AS THE ANSWER COMES OUT EVENLY. ESPECIALLY WHEN THERE IS A GOOD MOVE NOT TOO FAR AWAY, YOU SHOULD CONSIDER DIVISION AS A WAY TO MAKE THE RIGHT NUMBER. THIS EXPRESSION IS A TYPICAL EXAMPLE: (1+2)/3, WHICH WOULD HAVE RESULTED IN A TOWN! THEN YOU WOULD HAVE BEEN AT 60 WHILE I WOULD HAVE ENDED UP AT 40.

BOB IS AT 54 -- THE COMPUTER IS AT 40

IT'S MY TURN.. SPINNERS ARE: 2 4 2
MY EXPRESSION IS: 2+(4/2)
WHICH GIVES ME 4
I TOOK A SHORTCUT
I BUMPED YOU
THE COMPUTER IS AT 54 -- BOB IS AT 40

IT'S YOUR TURN.. SPINNERS ARE: 1 2 2
WHAT ARITHMETIC EXPRESSION DID YOU FORM?
=> 1+2*2
WHAT NUMBER DOES THAT GIVE YOU?
=> 5
Move ranks 11 which is POOR
The ISSUES are: BUMP SHORTCUT ORDER STRATEGY

YOU DON'T SEEM TO BE BUMPTING VERY MUCH. BUMPS ARE HARD TO GET BUT ARE USUALLY A GOOD IDEA. ONE GOOD EXAMPLE WOULD BE THE EXPRESSION: (1*2)+2, WHICH WOULD GIVE YOU A SHORTCUT AND A BUMP!! SO YOU COULD HAVE BEEN AT 54 WHILE I WOULD HAVE ENEDED UP AT 40.

BOB IS AT 45 -- THE COMPUTER IS AT 54

IT'S MY TURN.. SPINNERS ARE: 1 2 1
MY EXPRESSION IS: (!+!)*2
WHICH GIVES ME 4
THE COMPUTER IS AT 58 -- BOB IS AT 45

IT'S YOUR TURN.. SPINNERS ARE: 2 0 7
WHAT ARITHMETIC EXPRESSION DID YOU FORM?
=> 2*0+7
WHAT NUMBER DOES THAT GIVE YOU?
=> 14

The WEST system also contains a simple arithmetic expression diagnostician which looks for mixed up precedence.

MULTIPLICATION IS DONE BEFORE ADDITION SO 2*0+7 IS EQUAL TO (2*0)+7 NOT 2*(0+7). WOULD YOU LIKE TO CHANGE YOUR EXPRESSION?
=> YES

SPINNERS ARE: 2 0 7
WHAT ARITHMETIC EXPRESSION DID YOU FORM?
=> 0+2*7
WHAT NUMBER DOES THAT GIVE YOU?
=> 14
Move ranks 3 which is GOOD
BOB IS AT 59 -- THE COMPUTER IS AT 58

IT'S MY TURN.. SPINNERS ARE: 3 1 3
MY EXPRESSION IS: (1+3)*3
WHICH GIVES ME 12
THE COMPUTER IS AT 70 -- BOB IS AT 59
I WIN.

THE COMPUTER HAS WON 382, LOST 26 AND TIED 98
BOB HAS WON 0, LOST 1 AND TIED 0

THANKS FOR PLAYING.

III. TECHNICAL DETAILS

The overall operation of the modelling/tutoring system can be briefly restated as follows. Through the course of the game, the student's behavior is compared with that of the Expert with respect to a collection of issues. From these comparisons, a model of the student's performance is built by the Modeller. When the student makes a less-than-optimal move, the Tutor uses the model together with the performance of the Expert to determine an issue in which the student is weak and which would have resulted in a better move. The Tutor then explains the issue to the student using the better move which illustrates it as an example. In this section we shall describe the issues, the Expert, the model, the Modeller and the Tutor which were used to generate the tutorial behavior manifested in the protocol.

The Issues

The issues define those aspects of the environment which are abstracted into the model and monitored by the tutoring module. They provided the organizing concepts which coordinate the activities of the modeller and the tutor. The issues currently addressed are:

Issue	Comment
ORDER of spinners	the spinners don't have to be used in any particular order.
PARENtheses	the use of parentheses is allowed and is frequently valuable.
BACKWARDS	if the result of an expression is negative the player moves backwards which can sometimes lead to a special move.
special moves	trying for TOWNs, BUMPs, SHORTCUTs is part of a

good strategy.

MINUS	subtraction is legal and often useful.
DIVIDE	division is legal and sometimes useful.
PATTERN	the operations can be used in any order, i.e. more than a small number of move patterns should be used.
STRATEGY	a strategy for looking for moves should be used, and alternative moves should be considered.

Each issue is defined by three subroutines (called procedural specialists): (1) a Recognizer which determines whether a move exhibits the issue; (2) an Evaluator which looks at a student model and determines whether the student is weak in the issue; and (3) a Speakers which generates explanatory English about the issue. The Recognizers are used by the Modeller to update the model on each turn and by the Tutor to determine if there are better moves which the student could have made which exhibit the issue. The Evaluators are used by the tutor to evaluate the student model in order to provide a set of possible student weaknesses. Speakers are used by the Tutor to explain the issue to the student, (e.g. "I notice that you seldom move backwards"). The intent and operation of each of these specialists will be described further within the framework of the overall system.

The Expert

The "Expert" module generates and evaluates the set of moves possible in a given situation. For West, the number of possible expressions (values) for each turn is small enough that the Expert can generate all of them. Each of the different values is then simulated to find the ending positions of both the player and his opponent (remember that a player's move can "bump" his opponent). In the evaluation strategy used by the Expert, the "goodness" of a move is the difference between the player's final position and his opponent's final position (called the "delta"). The Expert determines the list of legal moves (ordered from largest to smallest delta). When it is the computer's turn, the Expert need only determine the optimal move. When it is the student's turn, the Expert generates the entire move space. This allows the student's move to be judged relative to the other possible moves that he could have made. As we shall see, the move space is used by both the Modeller and the Tutor.

The Model

The student model is a record of the student's past performance which provides the Tutor with information which is useful in determining what to say. The model consists of a cumulative structural history of how the student has performed on the issues relative to the performance of the Expert. The structural model which was built by the West system during the protocol is given in Figure 3 and illustrates its various components.

In addition to the cumulative structural model, the system also maintains a history list which has a complete temporal record of the student's session. This includes for each move, the spinners, the expression entered by the student, the results of the move (bumps, towns, etc.) and the final position. The information provided by the history list is needed, for example, to check the recent moves made by the student.

The Modeller

The task of the Modeller is to construct and maintain the structural model. Using the list of legal moves generated by the Expert, the Modeller first determines an overall "quality" of the student's move. The quality of the move is a rough classification of the move (as BEST, GOOD, FAIR or POOR) depending on how many better moves could have been made.* Each of the issue Recognizers is then invoked to update a particular portion of the model. Each Recognizer uses the set of better moves to judge the student's move with respect to its particular issue.

An Issue Recognizer

Since the major part of the Modeller's work is done by the individual Recognizers for each issue, we will describe in detail the operation of one such Recognizer, the Pattern Recognizer.** This example will provide a good overall view of the tasks and techniques for the other Recognizers.

The Pattern Recognizer is concerned with the form of the expression underlying a particular move. A move is classified into one of 16 possible patterns according to the operations used in the expression and the order in which they are performed.

---

*"Better" is with respect to the Expert's evaluation procedure (strategy) which is to maximize delta. See below for a discussion of the possibility of varying strategies.

**See Brown et al [1975] for a complete description of the Recognizers.

Figure 3: Snapshot of a student model

PATTERN	BEST	GOOD	FAIR	POOR	MISSED/BEST
(A+B)-C	0	0	0	0	3
(A*B)/C	0	0	0	0	3
(A+B)*C	0	0	0	0	3
(A*B)+C	0	2	3	2	2
(A+B)/C	0	0	0	0	2
(A*B)/C	0	0	0	0	1
A-(B+C)	0	0	0	0	1
A/(B+C)	0	0	[illegible]	0	2
A-(B*C)	0	0	0	0	1
A+(B/C)	0	0	0	0	1
A*(B-C)	0	0	0	0	1
A-(B/C)	0	0	0	0	1
(A-B)/C	0	0	0	0	2
A/(B-C)	0	0	0	0	2
(A/B)-C	0	0	0	0	1
TOTALS:	0	2	3	2	

The pattern section profiles the student's use and non-use of each of the 16 possible move patterns. The rows indicate the number of times the pattern was used for a move of each quality. The MISSED/BEST column indicates the number of times the pattern occurred as one of the optimal moves.

The totals provide an overall view of the strength of the player.

RANK:	1	3	9	11	13
NUMBER:	0	2	1	2	1

The ranking section gives the distribution of how the student's moves compared to an expert's. The RANK of a move indicates how many better moves there were. The NUMBER gives how many times that RANK occurred.

ORDER INFORMATION:

	ORIG	REV	LMS	SML	OTHER
GOOD:	1	0	0	1	0
POOR:	4	0	0	0	1

The order section profiles the order in which the spinners were used in the student's move. The orders which are considered are: ORIG, same as presented on the spinners; REV, reverse of spinners; LMS, decreasing order by size; SML, increasing order by size; and OTHER, none of the above. The subfields indicate the number of times the order was used when the quality of the move was GOOD (BEST or GOOD in pattern section) and POOR (FAIR or POOR in Pattern section).

DIRECTION INFORMATION:

	FORWARD	BACKWARD
GOOD:	2	0
POOR:	5	0
WAS/BEST:	7	0

The Direction section records the number of times the student's expression resulted in an initial move FORWARD (and BACKWARD) when the quality of the move was GOOD (or POOR). The WAS/BEST field indicates the directions of the optimal moves.

PARENTHESES: NECESSARY 0 OTHER 0 NONE 7

*The parenthesis section profiles the student's use of parentheses by noting the number of NECESSARY uses of parentheses as in (A+B)*C, the number of OTHER uses of parentheses and the number of times no parentheses were used.*

SPECIAL MOVES:

	TOWN	BUMP	SHORTCUT
TOOK:	0	0	0
WAS/BEST:	2	2	3

The Special moves section maintains for each of the special moves, TOWNs, BUMPs, and SHORTCUTs, how many times the student used that type of move (TOOK) and how many times the optimal move used it (WAS/BEST).

STRATEGIES:

SPECIAL	MAXDELTA	MAXNUMB	ENDGÁME	OTHER
0	0	1	0	6

The Strategy section keeps track of possible strategies the student may be using. The strategies are: SPECIAL, land me on a special move; MAXDELTA, maximize the difference between your position and your opponent's; MAXNUMB, make the largest number; and ENDGAME, land on 70. The counters indicate the number of student moves which were optimal under the corresponding strategy. OTHER keeps account of the moves which were not optimal under any of these strategies.

The model contains, for each pattern, the number of times the pattern was used, subclassified by the quality of the move. The Pattern section of the model provides a profile of the student's use of each pattern and identifies overused patterns (e.g. those which were used on FAIR or POOR moves). For example, it can be seen from Figure 3 that Bob overused the pattern A*B+C in the protocol.

In addition to information about what the student did, the Pattern Recognizer also maintains a record of what the student did not do! In particular, for those moves in which the student's move was not optimal the Pattern Recognizer increments the MISSED/BEST field for all of the patterns which could have given an optimal move. This information points out potential weak areas by indicating those patterns that the student did not use when he should have. In general, information about what the student could have done but didn't is very important as it must be used to avoid criticizing the student about issues which were never to his advantage to use.

The Tutor

The Tutor is responsible for deciding what to say and when to say it. Within the "issue and examples" paradigm, the range of possible "whats" is determined by the issues that are defined by the authors of the system. Exactly which issue and when it should be mentioned are determined by the student's behavior and a particular tutoring strategy. When the student makes a less than optimal move, the Tutor recognizes the event as an opportunity to generate advice. The Tutor calls the issue Evaluators (described in the next section) to determine the issues on which the student is weak. The list of weak issues constitutes the things the Tutor would like to tell the student about. However, just because a student is weak in something doesn't mean that this is the time to tell him about it. The student will only be interested if by using the issue he could have done better. Hence the Tutor uses the list of moves generated by the Expert, together with the Issue Recognizers, to determine if any of the better moves involve an issue in which the student is weak. The list of issues which result from this process can be thought of as the Tutor's hypotheses about why the student didn't make a better move.* For example, if the student has never used parentheses, and the best move requires parentheses, one such hypothesis is that he doesn't use parentheses. If the list of "tutorable" issues is empty, the tutor has nothing particular to say. It can make a general comment anyway ("I think I see a move which would get you farther...") but usually would not do so. If the list has more than one element, a choice between the issues must be made. At present, an ordered "issues list" is maintained which gives the relative importance of each issue. In a more complex domain, the issues could have, for example, a lattice structure where certain issues are prerequisite to other issues.

Once the Tutor has chosen an issue and an example,* the Speaker associated with the issue is invoked to provide feedback to the student. At present the Speakers are very simple. Each has three or four possible phrases for each of three or four parts of an explanatory paragraph. This implementation has the advantages of being easy to build and providing a reasonable variety of comments. The main limitation of such simplicity is that a Speaker which is not aware of the context in which it must "talk" (i.e. player positions, moves, etc.) must make very general comments (or risk making inappropriate comments) and hence miss chances for being particularly incisive.

An Issue Evaluator

The success of the Tutor depends critically on the ability of the Issue Evaluators to isolate the weaknesses of the student. As an example of the type of operation performed by the Issue Evaluators, we will describe the Pattern Evaluator. The Pattern Evaluator checks the student model (see Figure 3) to see if the student is varying the form of his move. The important factor is how the student's behavior compares with the Expert's. That is, how many times has the student used a given pattern when he could have done better with a different one. As mentioned earlier, the Pattern Recognizer classifies each move as one of 16 patterns depending on the operations and their order of operation. Thus if the student is always forming A+B*C, that field of the pattern section will have a large portion of the moves. Notice, however, that constant use of a single pattern does not necessarily indicate that the student is stuck. It may be the case that in these particular situations, the student made the best move. For this reason, the Evaluator

---

*The best such hypothesis is one which is exhibited by all of the better moves.

*The tutor also uses other strategies to limit its verbosity such as not hassling a student on an issue he has performed satisfactorily within the last three moves, not hassling the student two moves in a row, not hassling a poor player on a GOOD as opposed to BEST move. This type of tuning is critical to a smoothly operating system but is, at present, very ad hoc and will not be mentioned further in this paper.

looks at the non-optimal subfields of each pattern to determine how often the student used a form when it was not the optimal thing to do. The criteria the Pattern Evaluator uses to determine if the student is stuck in a pattern is: "Has the student used this pattern non-optimally more than 75% of the times that he has not used it optimally."

IV. DISCUSSION

When we began designing this system we faced uncertainty about what should go into a student model and how to guide the tutor into making insightful comments at relevant, and only relevant, times. Because of the lack of any comprehensive theory for how to grow and use student models or what constitutes useful tutorial comments, our system was designed so that it could be easily modified. That way, we could run subjects on the system, observe the system's behavior and the students' reactions, modify the system where necessary and eventually compare the system's behavior to that of human tutors (ourselves).

We would like to describe two techniques we found useful for evaluating successive versions of the West system. One method, which we used to determine the adequacy of the model, was to see if a human tutor, using just the model constructed by the Modeller and a given student's move, could determine the student's weakness. When the model contained so little or poorly structured information that a person could not generate reasonable comments, we saw no reason to believe a program should necessarily do so. The other method involved our playing West under a consistent but suboptimal strategy such as always using the spinners in the same order and never using parentheses. In such circumstances the tutor should comment on those (and only those) issues which we were purposefully avoiding.

Experimental Results

Although we have not yet conducted any major studies of how effective our tutoring system is, we have run several informal experiments. One of these consisted of running 18 student teachers in which each one used the system for at least one hour. Afterwards, each was asked to complete a questionnaire, with 12 complying. The following comments apply to this sample:

All but one subject received advice from the Tutor. Their comments about the Tutor were quite favorable. Nine subjects stated that the Tutor's comments were appropriate to what they were doing. Of the two who disagreed, one said that the Tutor was offering a strategy which he didn't feel he should follow because it would leave him "vulnerable to attack", an element of strategy not known to our current Expert. Eight out 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 Tutor manifested a good understanding of their weaknesses! One subject commented "I misunderstood a rule; the computer picked it up in the 2nd game."

We are quite encouraged by these results. Not only did the subjects sense the "intelligence" of the Tutor in knowing when to offer appropriate suggestions but they seemed to enjoy the Tutor's support. We, of course, realize that this data is highly subjective and are looking forward to conducting some more controlled experiments.

Extensions

While the present system has worked very well in experiments, there are several extensions to the paradigm worth mentioning. One deals with the problem of "changing the point of view" of the student model. The system evaluates a move based on its comparison to an Expert's move in the same situation. This Expert must use some strategy to decide which move is best. For example, is it better to get one farther or to be on a town? Whatever strategy the Expert uses (it currently uses the maximum delta strategy), it may not be the same strategy employed by the student. When this is the case, the student's moves won't be evaluated correctly using the Expert's strategy as a standard. If the reason for tutoring the student is not necessarily to teach him the Expert's notion of a good strategy, but instead to help him become aware of a wide range of issues, it might be beneficial to criticize the student within his own strategy. If we discover that the student is playing a coherent but different strategy (either by asking him or by noticing patterns in his Model*) the Modeller can re-synthesize the model using the history list and an Expert who simulates the particular student strategy. When the Expert correctly guesses and simulates the actual strategy, the resulting model will sharply indicate a better player. At this point, if we verbalize this strategy to the student, we can make him aware of it and hence willing to consider alternatives. This provides him with a goal in addition to arithmetic practice, i.e. he can experiment with strategies.

----

*The types of patterns in the model might be a large number of moves which are not optimal in any known strategy together with general strengths in other areas; i.e. when the student is making less than optimal moves which can't be explained by the issues.

A more general limitation stems from the issues and examples paradigm itself. At present the issues act like "demons" observing the student's activities, watching for situations in which they can point out something of interest. This technique is very good at taking advantage of the work that the student has done to point out interesting things. There is another dimension to tutoring which this technique does not capture. That is the notion of directing the student activities in a general direction in the hope of putting him in an interesting situation. For example, if a particular issue has never come up, we could bias the spinner values to try to make that issue arise. While our current system concentrates on bottom-up tutoring through issues it is clear to us that a general system must also include top down guidance.

Conclusions

The overall sense we had from building and experimenting with this system is that it is easy to talk about student models and yet surprisingly difficult to actually construct a system that can grow an insightful model of the student and then use this model in a sensitive way to tutor the student. The pedagogical value of drawing tutorial examples from the student's work seems beyond reproach; yet the intelligence the system must have to successfully act on its own is considerable. Constructing a tutor which constantly criticizes is relatively straightforward. The point is to make one that only interrupts when a skilled human tutor would and then generates a succinct remedial comment.

We feel that our West system and the general tutoring paradigm of "Issues and Examples" provides the beginning of a theory of how this can be accomplished. It also provides a glimpse of the technical problems which must be confronted in actually constructing an operational system that can grow and use student models in a versatile way.

Acknowledgements

We are especially indebted to Mark Miller who was deeply involved with the design and implementation of our initial tutorial version of West. We would also like to thank Dr. Cecily Resnick for getting us interested in West and giving us some initial ideas on structuring the student model. The design and implementation part of this project was funded, in part, by the Army Research Institute and Naval Personnel Research and Development Center. Further analysis of this system was funded in part under a Tri-Service (AFHRL, ARI, [illegible], and ARPA (HRRO) contract. The research was conducted as part of a student [illegible] discover better ways to construct and [illegible] structural models of the learner for various training applications such as au[illegible]ting training simulators with intelligent computer-based tutors.

REFERENCES

Brown, J.S., R.R. Burton, [illegible] Miller, J. DeKleer, S. Pureell, C. [illegible]ann, and R. Bobrow, Steps Toward a theoretical Foundation for Complex Knowledge-Based CAI, Final Report, August [illegible]

Resnick, [illegible]n "Computational Models of [illegible] for Computer Assisted Learning [illegible], Doctoral Dissertation, University of Illinois at Urbana-Champaign, 1975.