Tutoring an Entire Game with Dynamic Strategy Graphs: The Mixed-Initiative Sudoku Tutor

Allan Caine and Robin Cohen
David R. Cheriton School of Computer Science, University of Waterloo, Waterloo, Canada
Email: {adcaine, rcohen}@cs.uwaterloo.ca

Abstract— **In this paper, we develop a mixed-initiative intelligent tutor for the game of Sudoku called MITS. We begin by developing a characterization of the strategies used in Sudoku as the basis for teaching the student how to play. In order to reason about interaction with the student, we also introduce a student modeling component motivated by the mixed-initiative model of Fleming and Cohen that tracks what the student knows and understands. In contrast to other systems for tutoring games, we are able to interact with students to complete an entire game. This is achieved by retaining a model of acceptable next moves (called a strategy graph) and dynamically adjusting this model as the student plays the game. We present the overall architecture of the system followed by an explanation of the modules that encapsulate the rules of Sudoku. We also outline formulae for reasoning about interaction with the student that support mixed-initiative where either the system or the student can elect to direct the playing of the game. An implementation of the system is discussed, including examples of MITS interacting with students in order to tutor the game. To conclude, we discuss how this research is useful not only to gain insight into how to tutor students about strategy games but also to understand how to support mixed-initiative interaction in tutorial settings.**

Index Terms— **Intelligent Tutoring Systems, Mixed-Initiative Systems, Strategy Games, Student Modeling**

I. Introduction

Sudoku is a strategy game, developed in Japan, that is growing in popularity worldwide. The game consists of trying to fill in empty cells in a partially completed 9×9 grid[1], with a clear set of rules about the possible entries in a cell (using the numbers of 1 to 9 only) and the conflicts to avoid when completing cells (exactly one of any number in any one column, row and particular 3×3 blocks of the grid). A sample solved Sudoku puzzle is displayed in Figure 1 with the 3×3 grids delineated by darker lines.

In this paper we present MITS: **M**ixed-**I**nitiative Intelligent **T**utoring **S**ystem for Sudoku. The main contributions of our research will be to:

1) demonstrate how to tutor a student for an entire strategy game;

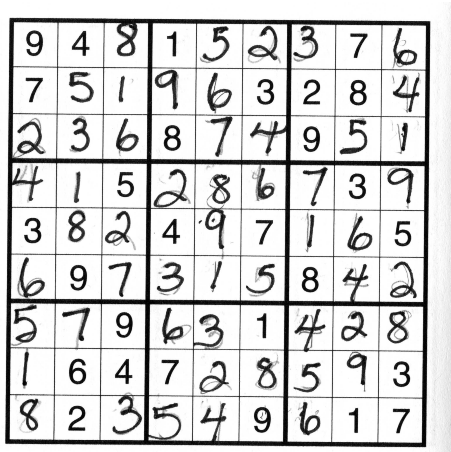


Figure 1. A solved Sudoku Puzzle.

2) allow for strategy graphs to be computed dynamically while tutoring students about a game;
3) present an architecture for mixed-initiative tutoring of games;
4) indicate how to adjust a framework for reasoning about interaction for mixed-initiative systems for the setting of intelligent tutoring; and
5) provide insight into what is required in order to tutor the game of Sudoku itself.

We begin with a discussion of related work in the areas of intelligent tutoring systems for games and mixed-initiative systems, looking more specifically at mixed-initiative tutoring and at models for reasoning about interaction in mixed-initiative settings. We argue that the choice of Sudoku is effective in order to design a system that can tutor an entire game and we provide details on how to characterize Sudoku game play for the purpose of tutoring.

Next we present the overall architecture of MITS and discuss how this extends the set of modules proposed in [2], in a critical way, to support mixed-initiative interaction and tutoring of an entire game board.

We then move on to discuss, at a high level, the processing algorithm underlying MITS. Our discussion clarifies the first phase used to build up information about the student, and the second phase where the system reasons about interacting with the student as the student attempts to complete a Sudoku game board. The details of the processing to capture mixed-initiative reasoning

This paper is based on "MITS: A Mixed-Initiative Intelligent Tutoring System for Sudoku," by A. Caine and R. Cohen, which appeared in AI'06, Québec City, Québec, Canada, June 2006 [1].

Financial assistance was received from the Natural Sciences and Engineering Research Council of Canada (NSERC).

[1]We focus on 9×9 Sudokus as found in newspapers although larger Sudokus are possible.

for the purposes of tutoring and to tutor the specifics of Sudoku are revealed in a separate section.

Details of the implemented system are discussed in a section that includes examples of interactions with students illustrating the proposed GUI interface. We conclude with a summary of the main contributions of the work contrasting with other research in the areas of intelligent tutoring and mixed-initiative systems.

This paper extends our earlier work presented in [1] in the following ways. Our assumptions about the game of Sudoku are clarified; this constitutes the starting point of our proposed approach for tutoring the game. The overall processing of the MITS system is characterized as a set of distinct phases including actions and iterations with terminating conditions. The strategies of Sudoku and the modules used within the MITS architecture for capturing these strategies are described in a detailed and comprehensive manner. The most important new design decisions introduced for both the architecture and implementation of the system are identified and discussed to emphasize the contributions of our research. We also include an extensive set of examples to illustrate the central processing algorithm covering various scenarios where tutoring may arise.

II. Related Work

According to Freedman [3], an intelligent tutoring system has four parts: a model of the domain, a model of the student, a model of the learning environment, and a teaching model. The model of the domain is what is being taught. The model of the student is a representation of the student by the tutor in the computer system. The model of the learning environment is essentially the user-interface. Finally, the teaching model is a representation of how the material is taught to the student. These important elements are retained in our proposed architecture for tutoring students in the game of Sudoku.

We also take as a starting point some research on the topic of teaching students about endgames of chess, the UMRAO system [2]. This system includes two major components: the Expert and the Tutor. The Expert is responsible for selecting the specific game board to be played and generating the strategy graph — a graph of all the possible next moves from a given game board position. Attached to each node is an explanation of the move. The Tutor uses the strategy graph to evaluate the moves made by the student, providing feedback and suggesting future moves to attempt. UMRAO incorporates as well a graphical interface that displays both the game board being considered and a the running feedback from the Tutor.

UMRAO is a valuable starting point for our research to develop a tutoring system for Sudoku, but it has a number of shortcomings that we attempt to address in our model, as follows:

1) it has no explicit student model. Instead, Gadwal *et al.* [2] claim that the student is modeled in terms of the strategy graph — the level of play demonstrated in that graph can serve to characterize the student as novice or expert;
2) UMRAO cannot switch its interpretation of the student's abilities during a tutoring session; as such it cannot adapt during the game play; and
3) the strategy graph that represents how to correctly play the game is computed off-line because of the great deal of time needed to compute it.

We elected to study a game that was less open-ended than chess, Sudoku, where the student's moves could be interpreted in terms of a small fixed number of possible strategies. We wanted to emphasize the opportunity for interaction during the tutoring session and to explore the circumstances under which interaction should take place, based on a modeling of the current state of the student. As such, we made sure that the student model is updated with each move to reflect the abilities of the student. In contrast to UMRAO, our strategy graph displays what is allowable as the next move in the game (as in Figure 4). We update the strategy graph only when the student has made an acceptable move; the graph reflects the student's progress in solving the Sudoku puzzle. While the student model reflects what the student knows and understands, the strategy graph reflects what the student has solved. In the end, the interaction provided in the system depends on the student and the state of the game, which is adjusted as the student becomes more skillful in playing the game.

Research in the field of mixed-initiative systems is aimed at determining how best to allow systems and users to work together in order to collaboratively solve problems. Some researchers have argued that intelligent tutoring is best addressed as a mixed-initiative activity, where students have some control over how the tutoring will proceed as in [4]. Others like [5], [6] have explored more specifically how to manage tutorial dialogues with students in a coherent manner. Still others have examined how to predict when the student will take the initiative; an example is [7]. A further key research question brought up in [8], [9] is how to model what the student is learning.

Our investigation of mixed-initiative tutoring is distinct. We are focused on how to reason about interacting with a student while attempting to tutor an entire game. What the student knows and is learning will be captured in simplified terms and in the context of the strategies underlying the game. The issues of managing an ongoing dialogue will also be less of a concern, because each new interaction will be selected based on the effort needed to progress in completing the game puzzle with the student.

We take as a starting point for the mixed-initiative modeling in MITS the research of Fleming and Cohen [10]. This work emphasizes the value of modeling the user when reasoning about interaction in a decision-theoretic framework that weighs the benefits and costs of interaction at any given point in time.

The first model of Fleming and Cohen [10] makes a distinction between what a user knows and what a user understands or can be made to understand through interaction. We elect to model the students in our Sudoku tutoring system according to these parameters. In particular, we will characterize the strategies required to solve Sudoku game puzzles and track the probability that the student knows these strategies, or is understanding them, through previous interaction with the tutor.

Another key element of the Fleming and Cohen model is the utility of an action chosen either by the system without the benefit of interaction with the user, or by the system once the user has had a chance to respond and to provide information. In MITS, we will adjust this factor to consider instead the utility of each move in the game with respect to completing the puzzle in the most efficient way.

A final component of the Fleming and Cohen work is the concept of cost. Interaction does generate possible bother to the user (as discussed in [11]). In MITS, the costs will be as a result of generating explanations during the course of tutoring.

It is important to note that the model of Fleming and Cohen was designed for collaborative problem solving environments, and, as such, it did not specifically examine the importance of interacting in order to enable the user to learn during the collaborative process. The importance of tracking what the student does not know and needs to be taught will be introduced within our model as part of the reasoning about interaction.

In the sections that follow, MITS will be presented in detail and the value of the new proposals for intelligent tutoring and mixed-initiative will be clarified.

III. Our Proposed Model: MITS

In this section, we present our system MITS, which will tutor a student playing the game of Sudoku using a mixed-initiative approach. We begin with a characterization of the game of Sudoku and its tutoring environment followed by a discussion of the proposed architecture of MITS and a description of the algorithm used by MITS to reason about interacting with the student.

A. Sudoku Strategies

Recall that the game of Sudoku is played beginning with a partially completed 9×9 grid of numbers (each of which is from 1 to 9) requiring the player to correctly fill the remaining values of the grid in accordance with the rules: that there is only one of any number in any column, row or delineated 3×3 block of the grid.

To tutor Sudoku, we first of all restrict ourselves to games where there is only one solution to the Sudoku puzzle. The aim is for the student to learn to play Sudoku sufficiently well so that the student will consistently choose the correct value for each empty cell (and make effective decisions about which cells to try to fill as play progresses), culminating in a completely filled and correct grid.

We make the design decision of focusing on the four basic strategies that a player can use in order to complete the open cells of the Sudoku puzzle, modeling to what extent the student knows and understands each of these four strategies towards the successful completion of the game. For the sake of brevity, we label these strategies $s_1 \ldots s_4$ with detailed examples of the use of these strategies presented in Section V.

- **Rows and Columns** s_1 We look at the numbers in the current rows and columns and determine what must be left by a process of elimination;
- **Blocks** s_2 This strategy is like s_1 except that we look at the 3×3 block to eliminate possibilities;
- **Pointing Pairs** s_3 In this strategy, we look for pairs of numbers[2] in the same row, column, or block which then cannot be possibilities in any other cells of that row, column, or block; and
- **Block-Line Reduction** s_4 In this strategy, we compare the values needed in a particular block to the values needed in any row or any column which intersects that block.

We are also careful to make a distinction between what are referred to as *playable* and *unplayable* cells. At any given stage of the game, it is important to reason from the known values of the other cells of the puzzle to determine for each cell whether it is possible to currently fill that cell with one value (i.e. the cell is playable) or whether there is more than one possible value for the cell (i.e. the cell is ambiguous and unplayable). For example, consider the 3×3 block g7-i9 of the Sudoku in Figure 2(a).[3] By virtue of the three 8's in cells c7, d9 and i6, there must be an 8 in cell g8. The three 5's in cells c9, e7 and g3 dictate that there is a 5 in cell i8. So, cells g8 and i8 are playable cells. Their values are shown in bold in Figure 2(b). It would be improper to play a 4 into cell i9 because that cell can be either a 4 or a 9 (shown as smaller numbers in that cell); cell i9 is unplayable. The same holds true of cell g7, which currently can be either 6 or 4. Understanding playable and unplayable cells enables the player to make moves in a correct and logical order.

B. Architecture of MITS

In Figure 3, we present the proposed architecture for MITS.

The *Puzzle Library* can be thought of as a database of Sudoku puzzles. An example of such a puzzle is given in Figure 2(a).

The *Expert* can be thought of as the game's referee. The Expert knows the game's rules and enforces them. The Expert has two main tasks: generating the Sudoku Strategy Graph (SSG), and generating the Sudoku Skill Matrix (SSM). As will be explained in Subsection III-D, the SSG indicates *what* the student can do; the SSM *how* the student can do it. The SSM will be discussed in more detail in Subsection III-E.

[2]It is also possible to have pointing triplets or quadruplets, but they occur less in practice than pointing pairs. So, we confine ourselves to pointing pairs in this paper.

[3]By convention, the columns of the Sudoku puzzle are labeled with letters with "a" on the left to "i" on the right. The rows are numbered from 1 to 9 with 1 at the top and 9 at the bottom.

1			2	3		4	7		
2		1					3	4	
3	3	6		1	2		5		
4	4	8			9	3			
5	2			7		8			6
6				6	1			5	8
7			8		5	9		1	3
8		4	9					7	
9			5	8		6	2		
	a	b	c	d	e	f	g	h	i

(a) Partially Completed Sudoku

1			2	3		4	7		
2		1					3	4	
3	3	6		1	2		5		
4	4	8			9	3			
5	2			7		8			6
6				6	1			5	8
7			8		5	9	6 4	1	3
8		4	9				**8**	7	**5**
9			5	8		6	2	9	9 4
	a	b	c	d	e	f	g	h	i

(b) Sudoku with some playable and unplayable cells

Figure 2. Sudoku Boards

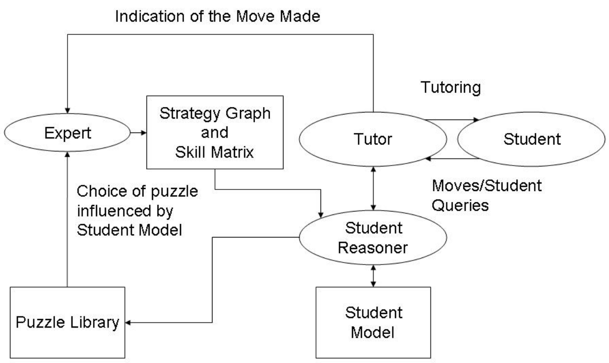


Figure 3. Our Proposed Model.

Once the Expert has generated the SSG and SSM, they are sent to the *Student Reasoner*. The Student Reasoner can be thought of as an educational consultant. It uses the *Student Model*, explained in Subsection III-F, to suggest to the *Tutor* the move to propose to the student. Once a correct move is made (i.e. the student has selected the correct value) the Tutor reports the move to the Expert, who updates the game. The Expert re-computes the SSG and SSM and sends them to the Student Reasoner in anticipation of the next move.

The Student Model keeps track of MITS's belief that the user understands the four strategies used to solve Sudoku puzzles in terms of acquiring the skill to apply them correctly. Using these probabilities, the Student Reasoner can assess the benefit and the cost associated with taking the initiative and tutoring a player regarding a particular move. Generally, if the benefits outweigh the costs, the player will be tutored. On the other hand, if the costs exceed or equal the benefits, MITS will invite the student to take the initiative and play independently.

When the student is taking the initiative, the Tutor will communicate the move made by the student to the Student Reasoner. The Expert will reply with information about the correctness or incorrectness of the move. When the student has attempted an incorrect move, MITS will re-take the initiative and begin a process of tutoring.

While MITS has some similarities with the UMRAO model of Gadwal *et al.* [2], there are also some important differences, introduced in order to enable the tutoring of an entire game and to support a mixed-initiative model of interaction between the student and the system.

In UMRAO, the strategy graph and the student model are essentially combined. In MITS, to support the mixed-initiative paradigm, it is imperative to be able to assess the student's ability to play Sudoku, independent of any consideration of what moves may be made next. We introduce a separate Student Model and also include a Student Reasoner in order to select the part of the puzzle to present to the student during tutoring. Since we maintain a dynamic strategy graph, this allows the Student Reasoner to constantly monitor the game for the best moves and strategies to tutor the student, in order to make the student's learning experience most effective; i.e. moves that draw the student closest to successful completion of the game. The Tutor is relieved of the responsibility for reading the strategy graph and is instead focused upon the dialogue with the student.

C. Reasoning about Interaction in MITS

Because we are using a mixed-initiative paradigm, we have the problem that the Tutor needs to know when to intervene and offer help, and when to permit the user to play without assistance. We argue that the model of Fleming and Cohen [10] can be easily incorporated into MITS, because we have focused our modeling of students in terms of four basic strategies of playing the game.

Generally, the Tutor will first take the initiative and strictly control the solving of the puzzle in an effort to impart some initial knowledge to the student; see Subsection III-F. As more moves are made by the student and witnessed by the Student Model, the Student Reasoner can develop a prediction of the success rate of the student in solving a Sudoku puzzle without assistance from the Tutor. As the probability that the student can solve the puzzle without assistance rises, the Student Reasoner will instruct the Tutor to allow the student to take the initiative. Even while the student is taking the initiative, the system will continue to monitor the student's performance. If the student's performance should degrade during unassisted play, the Tutor can resume assisting the student.

In accordance with the model of Fleming and Cohen [10], a cost-benefit analysis is conducted to determine if assistance should be given by the Tutor. In general, if the benefit of giving assistance exceeds the cost, the Tutor gives assistance.

$$B_{i,(x,y)} = [1 - P_{\text{UK}}(s_i)][P_{\text{UU}}(s_i) + (1 - P_{\text{UU}}(s_i))P_{\text{UMU}}(s_i)]\Delta U_{(x,y)} \tag{1}$$

$$P_{\text{UK}}(s_i) = \frac{\text{\# of times strategy } s_i \text{ was used correctly}}{\text{\# of times strategy } s_i \text{ was used}} \tag{2}$$

$$P_{\text{UU}}(s_i) = \frac{\text{\# of times strategy } s_i \text{ was explained and understood}}{\text{\# of times strategy } s_i \text{ was explained}} \tag{3}$$

$$P_{\text{UMU}}(s_i) = \frac{\text{\# of times } s_i \text{ was explained in detail and understood}}{\text{\# of times } s_i \text{ was explained in detail.}} \tag{4}$$

$$\Delta U_{(x,y)} = \frac{\text{\# of cells known unambiguously after move} - \text{\# of cells known unambiguously before move}}{81} \tag{5}$$

$$C_{i,(x,y)} = C_{i,(x,y)\,\text{simple explanation}} + (1 - P_{\text{UU}}(s_i))C_{i,(x,y)\,\text{detailed explanation}} \tag{6}$$

$$W_{i,(x,y)} = B_{i,(x,y)} - C_{i,(x,y)} \tag{7}$$

The benefit of giving assistance for a move into cell (x, y) using strategy i is given by Equation (1) where an explanation of the variables used in Equation (1) follows.

While play is underway, the Student Reasoner records the answers provided by the student in the empty cells. For each cell, one of the strategies would have been employed by the student. Consequently, the student model can keep simple tallies per Equation (2) to (4) for $i \in 1 \ldots 4$.

As given in Equation (2), $P_{\text{UK}}(s_i)$ is the probability that the student knows strategy i. The use of the factor $1 - P_{\text{UK}}(s_i)$ in Equation (1) is opposite to the use in [10]: the benefit of giving advice is inversely related to the probability that the student already knows strategy i.

Furthermore, as in Equation (3), $P_{\text{UU}}(s_i)$ is the probability that the student would understand the advice given for strategy s_i and $P_{\text{UMU}}(s_i)$ in Equation (4) is the probability that the student could be *made* to understand the strategy s_i.

At the outset, MITS does not know if a student would understand the advice given or if it would be able to *make* the student understand the advice given. So, these probabilities are initialized to 0.5. These probabilities would be adjusted, up and down, with actual experience as the student interacts with MITS and makes correct or incorrect moves. As illustrated in further detail in Section IV, explaining a strategy occurs when MITS takes the initiative to direct a user to fill a particular cell and a detailed explanation occurs when the student is provided with additional hints for filling a cell.

The change in utility that a particular cell, (x, y), contributes to the game is computed by Equation (5). Simply put, ΔU is a measurement of the extent to which a particular move advances the game towards the solution if $\Delta U > 0$, or the extent of a digression from the solution if $\Delta U < 0$. It is also possible for $\Delta U = 0$, which means that the move failed to disambiguate any other cells. In brief, Equation (1) suggests that: i) it is beneficial to interact if the student lacks knowledge and either would understand or could be made to understand the strategy that needs to be explained; and ii) it is more beneficial to interact about moves that resolve more of the game board.

The first part of Equation (1) in the first set of square brackets takes into account the expectation that the player understands the strategy. All other things being equal, the benefit is inversely related to $P_{\text{UK}}(s_i)$. For example, suppose that MITS believes with complete certainty that the player understands strategy s_i, which implies that $P_{\text{UK}}(s_i) = 1$. Regardless of the other values in Equation (1), $B_{i,(x,y)} = 0$. Intuitively, this result makes sense. If the player understands the strategy and MITS is completely convinced of that fact, then there is no benefit in tutoring the student about the strategy, s_i, regardless of the cell (x, y).

The second part of Equation (1) in the second set of square brackets addresses MITS's expectation that the player can be tutored if the need should arise. All other things being equal, it is more beneficial to choose to tutor the player regarding a move where MITS believes that the student can understand an explanation or be made to understand an explanation.

The third and final part of Equation (1), $\Delta U_{(x,y)}$, takes into account the utility of the move. All things being equal, MITS should choose to tutor the student regarding a move which is most beneficial in terms of completing the game. In other words, Equation (1) balances three factors:

- MITS's belief that the student understands the strategy underlying the move in cell (x, y);
- MITS's belief that the student can be tutored regarding the move in cell (x, y); and
- the strategic importance of the move in cell (x, y).

Cost would be measured as follows according to Equation (6) for some strategy $i \in 1 \ldots 4$. The costs $C_{i,(x,y)\,\text{simple}}$ and $C_{i,(x,y)\,\text{detailed}}$ would be each set on a scale $(0, 1]$ and would be in proportion to the difficulty of explaining the strategy to the user. Some of the four strategies are easier to explain than others. The first factor, $C_{i,(x,y)\,\text{simple}}$, is the basic cost involved in offering an explanation of the user. The second factor, $(1 - P_{\text{UU}}(s_i))C_{i,(x,y)\,\text{detailed explanation}}$, relates to the cost of a detailed explanation weighted by MITS's expectation that the detailed explanation will have to be given. Note that when the probability that the user understands is high, the cost of a detailed explanation is lower than when the probability that the user understands is low.

Equation (7) is a measure of *net worth*: the difference between the benefit and the cost. In general, if the Student Reasoner can find a move into cell (x, y) and a corresponding strategy i such that $W_{i,(x,y)} > 0$, then MITS will take the initiative and tutor the student in regards to move (m_x, m_y) and strategy i, where (m_x, m_y) refers to filling cell (x, y) with its value. Since there may be more than one move and corresponding strategy satisfying the condition $W_{i,(x,y)} > 0$, MITS makes its choice such that $(i, (x, y)) = \text{argmax}_{\hat{\imath},\hat{x},\hat{y}} W_{\hat{\imath},(\hat{x},\hat{y})}$.

On the other hand, if $\forall\, i, x, y\ W_{i,(x,y)} \leq 0$, then turning over the initiative to the student is indicated; the student should be allowed to play independently and without assistance.

So, under this approach, an explanation would be given if $B_{i,(x,y)} > C_{i,(x,y)}$ for some playable move (m_x, m_y) using strategy i. However, because of the manner in which we have defined utility, U, we must make an exception to the ordinary cost-benefit rule whenever $\Delta U_{(x,y)} < 0$ for some move (m_x, m_y) made by the student during unassisted play. If $\Delta U_{(x,y)} < 0$, then the benefit calculated according to Equation (1) will be negative. Since the cost is always non-negative, the cost-benefit rule is never met under these circumstances. Yet, whenever $\Delta U_{(x,y)} < 0$, the student has committed an error and MITS must intervene to correct the improper play. We discuss in Section VI why it may not be advisable to allow students to continue to play after making a wrong move in order to effectively tutor the game of Sudoku.

The ultimate goal of the student model is $B_{i,(x,y)} \leq C_{i,(x,y)}$ for every plausible move (m_x, m_y) and strategy i from the current board state. This means that the student is playing Sudoku independently. Yet, we cannot uncompromisingly apply the rule $\Delta U_{(x,y)} > 0$ and $B_{i,(x,y)} > C_{i,(x,y)}\ \forall i, (x, y)$ as a condition of giving advice. Consider that case where the student *wants* advice when the student is playing independently. For example, the student may be playing a puzzle at a high level of difficulty and has managed to "get stuck." In this case, the tutor computes $\text{argmax}_{i,\hat{x},\hat{y}} W_{\hat{\imath},(\hat{x},\hat{y})}$, in order to determine the most worthwhile move, (m_x, m_y), using strategy i to discuss with the user. An example of a student requesting help is displayed in Section V-B.1.

D. Sudoku Strategy Graphs

Sudoku Strategy Graphs (SSG's) are straightforward to read. A typical strategy graph is given in Figure 4(a). A cell with a single large number means that the cell was either an initial clue or it has been subsequently and correctly solved out by the student. A cell with one small number means that the cell is unambiguous, and, for the current board state, the cell can be solved out. If a cell in the SSG has two or more small numbers, the cell is ambiguous. The numbers appearing in the cell are the current possibilities, but the actual answer is unknown. A student who attempts to solve out an ambiguous cell is in fact committing an error.

Suppose a player inserts an 8 in square e5 of Figure 4(a). Prior to that move, there were 37 unambiguous cells. After making that move and with reference to Figure 4(b), there are 39 unambiguous cells, which implies that $\Delta U_{(\text{e},5)} = \frac{39-37}{81} = 0.025$.

As well, an SSG can become contradicted. Suppose a player plays "1" in cell i8 from the board state in Figure 4(b). The strategy graph that would result is illustrated in Figure 4(c). This play is improper because "1" is not a possibility for cell i8, because a "1" already appears in cell f8. Before the move is made, there are 39 unambiguous cells; afterwards, there are 34; so then $\Delta U_{(\text{i},8)} = \frac{34-39}{81} = -0.062$. So, whenever $\Delta U_{(x,y)} < 0$ the student requires assistance.

E. Sudoku Skill Matrix

It is not enough for the Expert to simply draw up the SSG. It must also compute the Sudoku Skill Matrix (SSM). The SSM is a 9×9 matrix each entry of which is an integer from 0 to 4. The numbers 1 to 4 indicate the strategy (or skill) that the student must use to solve the given position on the Sudoku board. If the Expert assigns a zero to any entry of the SSM, the zero signifies that either the cell has already been solved out, or that the cell is ambiguous and any attempt to solve the cell would be premature. The SSM related to Figure 4(a) is given in Figure 5.

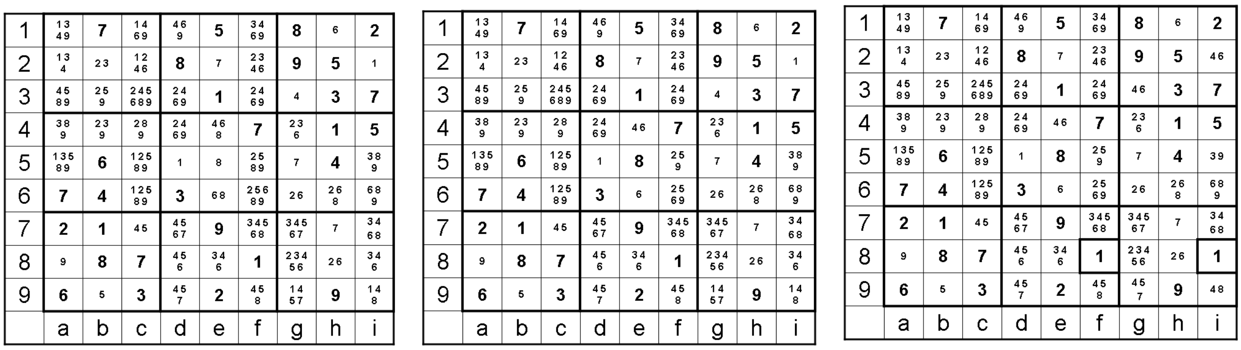
1	0	0	0	0	0	0	0	2	0
2	0	0	0	0	4	0	0	0	4
3	0	0	0	0	0	0	4	0	0
4	0	0	0	0	0	0	0	0	0
5	0	0	0	4	1	0	1	0	0
6	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	1	0
8	4	0	0	0	0	0	0	0	0
9	0	1	0	0	0	0	0	0	0
	a	b	c	d	e	f	g	h	i

Figure 5. A Sudoku Skill Matrix for the SSG in Figure 4(a).

Potentially, there might be more than one strategy available for solving out a cell in the Sudoku board. So, the programmer will need to provide for storage for additional integers in each cell. In the case of multiple strategies, we must consider two cases: the tutor has the initiative and the student has the initiative.

If the tutor has the initiative, then the strategy i will be selected such that $P_{\text{UK}}(s_i)$ is a minimum. Consequently, Equation (1) is maximized all other things being equal. The tutor will be focused upon the strategy that the student knows the least.

If the student has the initiative, the problem is more complex because the issue of *plan recognition* arises. If the student gives the right answer, the tutor does not know which strategy the student used. The tutor could query the user to find out, but queries following a *proper* move might be viewed as an annoyance to the user. We take the



(a) First board Position (b) SSG after an “8” is played in cell e5 (c) SSG after a “1” improperly played in i8

Figure 4. Sudoku Strategy Graphs.

view that it would be better to simply not trouble the user even though the probability $P_{\text{UK}}(s_i)$ will not be updated for that play. On the other hand, if the user supplies the wrong answer, the tutor can re-take the initiative and choose the strategy i such that $P_{\text{UK}}(s_i)$ is minimized over all plausible strategies. The tutor has moved the focus of discourse to the strategy least understood by the user.

F. Initializing the Student Model

There are two ways in which a student model can be initialized. The student can provide the parameters, or the system can develop the model during play. We take the position that the latter approach is best.

The user is modeled using 12 probabilities $P_{\text{UK}}(s_i)$, $P_{\text{UU}}(s_i)$, and $P_{\text{UMU}}(s_i)$ for $i \in 1 \dots 4$. We concede that the user *might* be able to provide the four probabilities $P_{\text{UK}}(s_i)$ provided that the user actually understands the strategies $s_1 \dots s_4$. The danger is that the user might believe incorrectly that they understand a strategy and overestimate one or more or the probabilities $P_{\text{UK}}(s_i)$.

Until a sufficiently large sample of moves is recorded by the student model, these probabilities cannot be used. However, the Student Reasoner keeps game play under strict tutor control selecting a variety of possible moves as the subject of discourse which exemplifies all four strategies $s_1 \dots s_4$. Once a sufficiently large sample of moves is obtained by the student model, the mixed initiative model can be brought on-line and used actively.

We have already suggested that the eight other probabilities $P_{\text{UU}}(s_i)$ and $P_{\text{UMU}}(s_i)$ should be all initialized to 0.5. MITS cannot anticipate how well its discourse with the student will fare. Likewise, since the student has yet to interact with MITS, the student cannot anticipate how well he or she will understand the MITS’ advice. So, it makes no logical sense for the student to provide those probabilities to the student model.

Unlike UMRAO, the Student Reasoner recommends the next puzzle to be played. In UMRAO, the Expert makes the decision. To account for this difference, the puzzles would be ranked by their level of difficulty. The difficulty level of a Sudoku puzzle is a function of the number of ambiguous cells from the first board state. The number of ambiguous cells can be determined from the puzzle’s first strategy graph.

MITS would start a new student off with a puzzle having the least degree of difficulty advancing the student upwards to the most difficult puzzle. When a puzzle is solved, it would be recorded by the student model to prevent the puzzle from being selected again. The student model would also store the level of difficulty of the last solved puzzle for reference in making future puzzle selections. Note that this procedure is conducted once per student.

G. The Algorithm

The MITS Algorithm is given in Figure 6. The first phase of the algorithm is the first repeat loop from lines 2 to 8. It involves establishing the Student Model. In this phase, MITS observes the student’s progress. However, the game is under the control of MITS.

The second phase is in the second repeat loop: lines 10 to 34. It commences whenever MITS is satisfied it has established a fair representation of the student in the student model. In line 13, MITS computes ΔU for all 81 cells. If any cell, (x, y), has the property that $\Delta U_{(x,y)} \geq 0$, it means that the cell is playable. So, in lines 11 to 15, MITS computes the benefit and cost of those cells as $B_{(x,y)}$ and $C_{(x,y)}$ using Equations (1) and (6) respectively.

At the if-statement at line 16, MITS decides if it should take the initiative. MITS takes the initiative whenever it finds a positive $W_{i,(x,y)}$ as computed per Equation (7); there is at least one move that is beneficial to discuss with the user. MITS will tutor the move with the greatest net worth. Note that when a move (m_x, m_y) is chosen to tutor the student, MITS will know the strategy that enables filling in the value of the chosen cell. This strategy will be suggested whenever MITS engages in detailed explanation to the student. If there is more than one possible strategy, MITS will be explaining the one that maximizes the value of $W_{i,(x,y)}$.

The student takes the initiative in line 20 whenever $W_{i,(x,y)} \leq 0 \, \forall \, i, x, y$, for under these circumstances there is no benefit in discussing any of the moves with the user. When the player has the initiative, MITS must anticipate four possible scenarios:

- **The student requests assistance**: In this case as in line 23, MITS must re-take the initiative and begin tutoring the student regarding move (m_x, m_y) such that $W_{i,(x,y)}$ is maximum.
- **The student does not request assistance, and**:
  - **The student makes a play into an unplayable cell**: An unplayable move is characterized by $\Delta U_{(x,y)} < 0$; see line 26. MITS must intervene and prohibit the move; otherwise, the game cannot be concluded successfully.
  - **The student makes a play into a playable cell, and**:
    - **Inserts the wrong number**: In this case per line 28, the student must be tutored on the correct play. The student's probabilities must be penalized to reflect the incorrect play.
    - **Inserts the correct number**: As in line 31, the student has made the correct play into a playable cell. The student is rewarded for the correct play by having the probabilities associated with that play increased.

After one of these four cases have been dealt with by MITS, MITS loops back to line 10 unless the game is concluded. MITS re-computes the $\Delta U_{(x,y)}$'s, $B_{(x,y)}$'s, and $C_{(x,y)}$'s in lines 13 to 15. Since the probabilities may have been changed in lines 30 or 32, it is possible that the condition in line 16 may change from the last iteration. To put it another way, MITS may retake the initiative if the player played poorly in the last iteration (line 30); MITS may relinquish the initiative to the player if the player played well in the last iteration (line 32).

1 /* PHASE ONE */;
2 **repeat**
3   **if** *game over* **then**
4   Select a new game from the puzzle library
5   Select move for the student to attempt;
6   Build up the Student Model;
7   /* No reasoning about mixed initiative */
8 **until** *sufficient statistics on student* ;
9 /* PHASE TWO */;
10 **repeat**
11   **for** $x \leftarrow 1$ **to** 9 **do**
12   **for** $y \leftarrow 1$ **to** 9 **do**
13   Compute $\Delta U_{(x,y)}$;
14   Compute $B_{i,(x,y)}$ and $C_{i,(x,y)}$;
15   /* for those i indicated by the SSM */;
16   **if** $\exists\, i, x, y \ni W_{i,(x,y)} > 0$ **then**
17   /* MITS Takes the initiative */;
18         $i, x, y \leftarrow \mathrm{argmax}_{\hat{\imath},\hat{x},\hat{y}} W_{\hat{\imath},(\hat{x},\hat{y})}$;
19   Tutor the move into cell (x, y);
20   **else**
21   /* Student has the initiative */;
22   /* $\forall i, x, y\ W_{i,(x,y)} \leq 0$ */;
23   **if** *student requests help* **then**
24             $i, x, y \leftarrow \mathrm{argmax}_{\hat{\imath},\hat{x},\hat{y}} W_{\hat{\imath},(\hat{x},\hat{y})}$;
25   Tutor the move into cell (x, y);
26   **else if** *student moves into cell* $(x, y) \ni \Delta U_{(x,y)} < 0$ **then**
27   MITS intervenes and prohibits
28   **else if** *student puts wrong number into* (x, y) **then**
29   Tutor student (x, y) to correct error;
30   Penalize student's probabilities;
31   **else**
32   /* Student has made a correct move */;
33   Reward the student's probabilities;
34 **until** *game concluded* ;

Figure 6. The MITS Algorithm

IV. GRAPHICAL USER INTERFACE

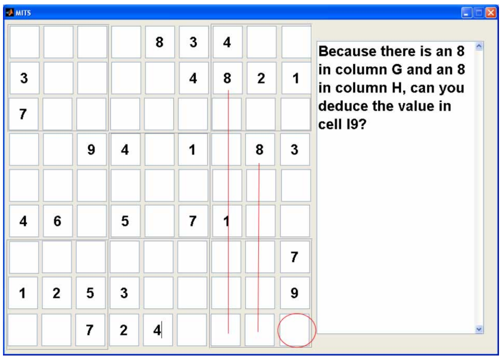
The Graphical User Interface is illustrated in Figure 7(a). We have opted for a simple GUI patterned after UMRAO. The playing board is to the left; the discourse window to the right. In Figure 7(a), the student model is being built and the Tutor has the initiative. The test concerns strategy s_4, Box-Line Reduction. Since there is an 8 in each of columns g and h and the columns intersect the block in the lower right, the user ought to be able to conclude that the value in cell i9 is an 8. We call this a detailed explanation (refer to Equation (4)) because the Tutor specifically names the other cells needed in making the logical deduction that i9 is "8.". Detailed explanation occurs during initial tutoring.

Assume that the user correctly plays 8 in cell i9. MITS continues to control the tutoring and directs the student to examine cell g7. This is a test of another variation of strategy s_4. Here, the student must recognize that there exists a 2 in each of rows 8 and 9, *and* a 2 in column h. So, it follows that g7 is 2. The Tutor is trying to ascertain if the student can use what was learned in board Figure 7(a) to solve cell g7 in board Figure 7(b). We call this a simple explanation (refer to Equation (3)), because the Tutor merely indicates where the move ought to be made, but does not explain the logic. If the student cannot determine the value of the cell, then the Tutor will resort to a detailed explanation similar to Figure 7(a).

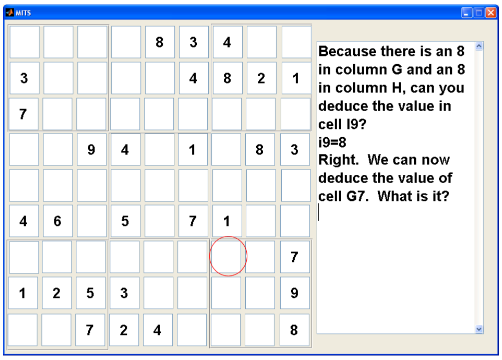
V. SAMPLE SESSION

In this section, we wish to demonstrate MITS under a number of scenarios. The scenarios we wish to consider are:

1) MITS takes the initiative, and
   a) Illustrates Strategy s_1 — rows and columns
   b) Illustrates Strategy s_2 — Blocks
   c) Illustrates Strategy s_3 — Pointing Pairs
   d) Illustrates Strategy s_4 — Block-Line Reduction



(a) Sudoku GUI — Move #1 Detailed Explanation



(b) Sudoku — Move #2 Simple Explanation

Figure 7. The MITS Graphical User Interface.

2) The player takes the initiative, and
   a) The player requests assistance;
   b) The player makes a play into an unplayable cell;
   c) The player makes a play into a playable cell but enters the wrong number into that cell;
   d) The player makes a play into a playable cell and enters the correct number into that cell.

For the purposes of our illustration, we assume that the $\Delta U_{(x,y)}$'s, $B_{(x,y)}$'s, and $C_{(x,y)}$'s in lines[4] 13 to 15 have been computed.

The examples presented below would all be cast in the GUI presented in Section IV. The interaction from MITS would appear in the column on the right hand side of the screen and the student would interact by clicking on certain empty cells and filling in certain values. In order to reduce the display of screenshots below, we simply show the initial game board that would appear and convert the student's turns into actual responses within a dialogue with MITS.

A. *MITS takes the initiative*

Recall that this situation is characterized by $\exists\, i, x, y \ni W_{i,(x,y)} > 0$ for some playable move (m_x, m_y); see line 16. MITS focuses on encouraging the student to put the correct value into a cell using the SSM in order to determine the strategy at hand. See Section VI for some thoughts on how we may also encourage students to acquire a deeper understanding of the strategies of Sudoku.

[4]For the sake of brevity in this section, whenever we refer to a line number, we are referring to a line number with respect to Figure 6.

1) MITS Illustrates Strategy s_1: Strategy s_1 is the row and column strategy. From Figure 8, it is possible to determine the position of the 9 in column g. The 9's in i7, h5 and a1 eliminate all possibilities other than g2; so g2 = 9.

1	**9**	2		8	5				
2	7		3		2			6	
3	8		1				4	5	
4	6					1	8		
5		1	2	4		3	7	**9**	
6			7	2					5
7		4	5				5		**9**
8		9			3		2		7
9					4	8		1	3
	a	b	c	d	e	f	g	h	i

Figure 8. Sudoku for row and column strategy

Consider a possible interaction between MITS and the Player as depicted in Figure 9. The notation `M>>` means MITS is displaying a message on the GUI; `P>>` means the player is interacting with the GUI. In this example, the user initially gives the wrong answer, g1; MITS reminds the player that there is already a 9 in row 1 and asks for another response. The player responds correctly.

```
M>> Where is the value 9 located in
column g?
P>> g1=9
M>> No, there is already a 9 in row 1
by a1=9. Try again.
P>> g2=9
M>> Correct!
```

Figure 9. Possible Interaction over rows and column strategy

2) MITS Illustrates Strategy s_2: This strategy is the blocks strategy where we try to determine a value by elimination. Suppose that we placed a 4 in c1 because of the 4 in b7, and a 5 in b2 because of a 5 in h3. This situation after these two moves is depicted in Figure 10. Then, by a process of elimination, the value in b3 must be 6 for it is the only value left which is still needed by the block a1-c3.

A possible interaction is given in Figure 11. Because of the simplicity of the situation, the student gives the correct answer at once.

3) MITS Illustrates Strategy s_3: To illustrate the pointing pairs strategy, s_3, consider Figure 12. In the 3×3 block from g4-i6, we need a digit 1. However, it is not clear if the 1 is in cell i4 or i5. On the other hand, if we consider the block from d4-f6, the 1 is in either e4 or f4 as denoted by the two tiny 1's. Those two 1's are the pointing pair, which indicate that there can be no other 1 on row 4. Consequently, the 1 in the g4-i6 block must be in cell i5 – not i4.

1	9	2	**4**	8	5				
2	7	**5**	3		2			6	
3	8		1				4	5	
4	6					1	8		
5		1	2	4		3	7	9	
6			7	2					5
7		4	5				5		9
8		9			3		2		7
9					4	8		1	3
	a	b	c	d	e	f	g	h	i

Figure 10. Sudoku for blocks strategy

```
M>> All of the cells in block a1:c3
are known except b3.
M>> What is the value of cell b3?
P>> b3=6
M>> Correct!
```

Figure 11. Interaction using the blocks strategy

A possible interaction with MITS is given in Figure 13. Note that this strategy is somewhat more subtle. MITS must lead up to it by first getting the student to locate the pointing pairs by asking where the 1's must go in block d4-f6. Only then can MITS ask the student where the 1 is located in block g4-i6.

4) MITS Illustrates Strategy s_4: In the block-line reduction strategy, we compare values needed in a particular block to the values needed in a particular row or column. Consider column b and block a1-c3 in Figure 14. Clearly, block a1-c3 needs a 4. On the other hand, column b does not need a 4; a 4 already appears in cell b7. So, by elimination, the 4 needed by block a1-c3 must appear somewhere other than column b; it must appear in cell c1.

A possible interaction over the block line reduction is given in Figure 15. Because of its simplicity, the player gives the correct answer at once.

B. The Player takes the initiative

The player takes the initiative whenever $\forall i, x, y \;\; W_{i,(x,y)} \leq 0$ as per line 20. The player takes the initiative because the foregoing conclusion does not indicate a need to tutor the player. However, while the player is taking the initiative, there are four possibilities as outlined below:

1) Requests Help: This situation arises whenever the student has been playing well, but the player desires a hint. This situation can be found on line 23.

In Figure 16, we show a Sudoku rated for advanced players. It is a difficult Sudoku because there are few open moves. So, a possible interaction with MITS where the player has the initiative is given in Figure 17. As it turns out, h7=2 by the block-line strategy.

2) Unplayable Move: Unlike Section V-A, we must consider the case where the player might make a move into a cell which is unplayable. By contrast, MITS will

1			7	6			5	4	8
2	9			6			1	2	3
3		5			4		8	7	6
4				4	1	1	8	5	
5		8		7	2	6		3	
6			1			5			
7	3	2			5			1	
8							2		5
9	6	7	5			2	3		
	a	b	c	d	e	f	g	h	i

Figure 12. Sudoku for pointing pairs strategy

```
M>> Where must the 1's go in block
d4-f6?
P>> e4=1 or f4=1
M>> Correct! So, where is the 1 in
block g4-i6?
P>> i5=1
M>> Correct!
```

Figure 13. Interaction for pointing pairs

1	9	2		8	5				
2	7		3		2			6	
3	8		1				4	5	
4	6					1	8		
5		1	2	4		3	7	9	
6			7	2					5
7		4	5				5		9
8		9			3		2		7
9					4	8		1	3
	a	b	c	d	e	f	g	h	i

Figure 14. Block Line Reduction

```
M>> Where is the 4 in block a1-c3?
P>> c1=4
M>> Correct!
```

Figure 15. Possible interaction over block line reduction

1		4		9	8				
2		9					2	4	
3	5		7		4	6			
4							3	9	1
5					3				
6	9	7	3						
7				8	7		1		4
8		3	2					8	
9					2	4		5	
	a	b	c	d	e	f	g	h	i

Figure 16. Advanced Sudoku

```
P>> help
M>> There is a move at cell h7
```

Figure 17. Interaction with MITS using help

never suggest that a player make a move into an unplayable cell.

The situation is illustrated in line 26. It it characterized in two ways: cell (x, y) is unplayable if $\Delta U_{(x,y)} < 0$ or if the SSM at cell (x, y) is zero[5]. MITS must intervene. Otherwise, if the move is allowed to stand, the game is guaranteed to end in a deadlock.

On the other hand, no penalty is possible. Since there is no strategy associated with an unplayable move, there is no strategy to penalize. Since the probabilities will not change, the $\Delta U_{(x,y)}$'s, $B_{(x,y)}$'s, and $C_{(x,y)}$'s won't change. Instead, MITS will simply suggest that the student try another move because the cell that the player tried is unplayable. The player retains control and takes the initiative on the next move.

Reconsider the advanced Sudoku in Figure 16. Now, in block a4-c6, it is certainly the case that the 1 is not in cells a4, b4, and c4. Yet, consider the interaction with MITS in Figure 18 where the player elects to play a 1 in cell a5.

```
P>> a5=1
M>> There is no open move in that
cell. Try again
```

Figure 18. Interaction where user makes an unplayable move

Note that MITS does nothing more than point out that no move exists. Furthermore, it does not matter that it may come to pass that a5=1, because MITS discourages guessing. However, since the benefits and costs of interaction have not changed, the player is invited to try again.

3) Playable Move — Wrong Entry: This situation can be found at line 28. Here, the student has found a move which is playable, but, unfortunately, enters the wrong number into the cell. In this case, MITS assumes that the strategy is not understood, penalizes the student's probabilities for that strategy, takes the initiative, and tutors the student. After the tutoring is complete, MITS may or may not return control back to the user depending upon the condition $\exists i, x, y \ni W_{i,(x,y)} > 0$. Since some of the student's probabilities were reduced, some of the $B_{(x,y)}$'s will increase. So, the foregoing condition might be made true in which case MITS would retain the initiative until the foregoing condition was made false again.

This result is intuitively correct. If the error is major in nature, then it ought to be the case that $\exists i, x, y \ni W_{i,(x,y)} > 0$; so, MITS should take the initiative on the next move. On the other hand, if the error is minor in nature such that in spite of a decrease in the player's probabilities it is still the case that $\forall i, x, y\ W_{i,(x,y)} \leq 0$, then MITS will not over-react to a minor error. The player will take the initiative on the next move in spite of the minor error. What constitutes a major or a minor error depends upon the extent to which the probabilities shift against the player when the player is penalized for the incorrect play.

Consider again the Sudoku in Figure 16. Further, let us suppose that the user recognizes that there is a play to cell f5, which there truly is. Suppose that due to a lapse in concentration, the user plays f5=8. The correct move is f5=9 by recognizing the three 9's in cells a6, h4, and d1. The interaction might be as in Figure 19.

```
P>> f5=8
M>> That's not quite right. Try again.
P>> f5=7
M>> Notice that a6=9, h4=9, and d1=9.
Try again.
P>> f5=9
M>> Correct. Since you had difficulty
with block-line, let me suggest the
next move. What is h7?
P>> h7=2
M>> Correct. Keep playing.
```

Figure 19. User makes a wrong play into a playable cell

Notice that when the user makes an incorrect response the second time, MITS gives an extra hint. Once the player does give the correct answer, the initiative shifts to MITS since it believes that a discussion to reinforce the ideas of the block-line strategy is worthwhile. So, it directs the player's attention to cell h7. We assume that the player answers correctly on the first attempt. Satisfied that the user understands the block-line strategy, MITS shifts the initiative back to the player and invites the player to play again.

4) Playable Move — Correct Entry: This situation, depicted in line 32, is the most desirable. The player has played correctly and independently into a playable cell. MITS rewards the student by increasing the player's probabilities for the strategy that was employed to solve that cell. Since some of the player's probabilities increase, some $B_{(x,y)}$'s will decrease. Consequently, the condition $\forall i, x, y\ W_{i,(x,y)} \leq 0$ remains true. The player retains the initiative on the next move. Intuitively, this result makes sense: if the player is playing correctly and independently, any intervention by MITS would be unnecessary and possibly unwelcome to the player.

Again, reconsider the Sudoku in Figure 16. For MITS, this is the easiest case to consider. It need only record the move and ask the player to play again. Suppose the player correctly plays f6=9. A possible interaction is given in Figure 20.

```
P>> f6=9
M>> Correct! Play again
```

Figure 20. Interaction where the user makes a correct play into a playable cell

VI. CONCLUSIONS AND FURTHER RESEARCH

In developing MITS, a system for tutoring a student about Sudoku, we have designed an architecture to

[5]To be clear, these conditions are equivalent: $\Delta U_{(x,y)} < 0$ iff the SSM at (x, y) is zero.

support mixed-initiative interaction during tutoring with a student being modeled and advised about an entire game. This is in contrast to other efforts to tutor games that are restricted to endgames only (eg. [2]). Using dynamic strategy graphs, we are able to model the student progressively during game play.

The domain of Sudoku is generally helpful for investigating mixed-initiative tutoring because there is an intuitive interpretation of the utility of a move in terms of the ability to disambiguate any open cells in the grid. We are then able to make use of this term of utility to critique the actions of the student leading to intervention from the tutor when the student lacks knowledge and is following paths that have low utility.

We have been able to introduce some innovative changes to the model of Fleming and Cohen for the design of mixed-initiative systems [10] in order to apply it to the problem of intelligent tutoring. Providing for a model of whether a user understands or can be made to understand when engaged in dialogue leads to a tutorial system that tracks the understanding of the student based in part on past attempts. In addition, the need to ensure that we are also enabling learning serves to adjust the decisions about interaction in the mixed-initiative model.

In contrast to other researchers focused on developing mixed-initiative tutoring systems [4]–[8] we specify how to integrate a definitive decision-theoretic model for reasoning about interaction with the student. In addition, we work directly from a student model that has been built up dynamically in previous sessions with the student. Within the context of Sudoku, we are able as well to condense the modeling of the student into a representation of the student's abilities with respect to the central strategies for playing the game.

There are several avenues for future research. In particular, developing a more sophisticated Student Reasoner and a more detailed student model would both be helpful in order to deliver more customized tutoring to the student. For instance, the Student Reasoner could identify patterns of difficulty in a student's strategy graph in order to predict values for the $P_{\text{UU}}(s_i)$ and $P_{\text{UMU}}(s_i)$ variables manipulated in the formulae. Another suggestion is to estimate more precisely the value of certain factors, such as the expected number of interactions to explain a given strategy by analyzing the student's current knowledge and past behavior. The suggestion of allowing for either very basic or more detailed commentary when advising a student is yet another area where more intelligent algorithms may be designed mapping a certain range of values of student modeling factors with a proposed level of detail for the interaction.

Note that we elected to have MITS interact with students whenever wrong moves were attempted (see Section V-B.1). We did not allow the student to continue to play a game with wrong moves because there are Sudoku puzzles where it may take a long number of moves before it is apparent that the game cannot be completed. While it would be possible to show the student at that what move long ago generated the difficulty, it would be challenging to progressively step through the various contexts that led from that move to the final deadlock.

Since we do inform students when they have made incorrect moves, it may be useful for future work to explore the possibility of pulling a student away from a puzzle entirely, if it is proving to be too challenging (i.e. a vast number of repeated wrong moves), replacing this with a somewhat simpler puzzle to tutor.

Another avenue for future work is to do a second pass with students, going over puzzles already completed in order to convey the essence of the basic strategies of Sudoku, to have students acquire this deep knowledge. Note that it would be challenging to simply "give" the students the strategies to begin with, because they are best understood with illustrative examples. We therefore decided to have the student experience the examples by playing the puzzles.

REFERENCES

[1] A. Caine and R. Cohen, "MITS: A mixed-initiative intelligent tutoring system for sudoku," *AI 2006*, June 2006.

[2] D. Gadwal, J. E. Greer, and G. I. McCalla, "Tutoring bishop-pawn endgames: an experiment in using knowledge-based chess as a domain for intelligent tutoring," *Applied Intelligence*, vol. 3, pp. 207 – 224, 1993.

[3] R. Freedman, "What is an intelligent tutoring system?" *Intelligence*, vol. 11(3), 2000.

[4] S. Carberry, "Discourse initiative: Its role in intelligent tutoring systems," *Working Notes of the AAAI Spring Symposium on Computational Models of Mixed Initiative Interaction*, pp. 10 – 15, 1997.

[5] F. Shah and M. Evens, "Student initiatives and tutor responses in a medical tutoring system," *Working Notes of the AAAI Spring Symposium on Computational Models of Mixed Initiative Interaction*, pp. 138 – 144, 1997.

[6] R. Freedman, "Degrees of mixed-initiative interaction in an intelligent tutoring system," in *AAAI 1997 Spring Symposium on Computational Model for Mixed-Initiative Interaction.*, 1997.

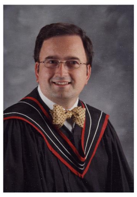
[7] J. Beck, P. Jia, J. Sison, and J. Mostow, "Predicting student help-request behavior in an intelligent tutor for reading," in *Proceedings of User Modeling 2003*, 2003.

[8] M. Manske and C. Conati, "Modelling learning in an educational game," in *Proceedings of AIED 2005*, 2005.

[9] A. Baena, M. Belmonte, and L. Mandow, "An intelligent tutor for a web-based chess course," in *Lecture Notes in Computer Science*, P. Brusilovsky, O. Stock, and C. Strapparava, Eds., vol. 1982/2000, Aug. 2000.

[10] M. Fleming and R. Cohen, "A user modeling approach to determining system initiative in mixed-initiative ai systems," *Proceedings of User Modeling 2001*, July 2001.

[11] R. Cohen, M. Cheng, and M. Fleming, "Why bother about bother: Is it worth it to ask the user?" *AAAI'05 Fall Symposium on Mixed-Initiative Problem-Solving Assistants*, 2005.



Allan Caine received his B.Sc. (Computer Science and Mathematics) from Luther College, University of Regina in 2002. In October 2005, he received his M.Math degree from the University of Waterloo. Currently, he is pursuing his Ph.D. at the University of Waterloo under the supervision of Richard Mann. He is affiliated with the Artificial Intelligence Research Group at the University. He researches computer vision focusing on the use of image transforms for image understanding. The research presented in this paper arose from work conducted for the course CS785, Intelligent computer Interfaces, in the Fall of 2005.

Robin Cohen is a Professor in the David R. Cheriton School of Computer Science at the University of Waterloo where she has been a faculty member since 1984. She conducts research in artificial intelligence in the areas of mixed-initiative systems, user modeling, intelligent interaction and multi-agent systems. She has served on the Advisory Board of User Modeling, Inc.; has acted as the program chair for UM'92 and AI'02 (the Canadian AI Conference); and has been on the program committees of numerous conferences and workshops in the artificial intelligence community.