A practical student model in an intelligent tutoring system

Conference Paper · February 1999

DOI: 10.1109/TAI.1999.809759 · Source: IEEE Xplore

2 authors:

Yujian Zhou
University of California, Berkeley

Martha W. Evens
Illinois Institute of Technology

Proceedings of the 11th IEEE International Conference on Tools with Artificial Intelligence. Chicago, IL, 1999, pp. 13-18

A Practical Student Model in an Intelligent Tutoring System

Yujian Zhou
Martha W. Evens

Department of CSAM, Illinois Inst. of Technology
10 W. 31st Street 236–SB, Chicago, IL 60616
zhouyuj@charlie.iit.edu, csevens@minna.cns.iit.edu
http://www.csam.iit.edu/~circsim
(312)567-5153 (312)567-5067 (fax)

Abstract

In this paper we consider two questions related to student modeling in an intelligent tutoring system: 1) What kind of student model should we build when we design a new system; 2) Should we divide the student model into different components depending on the information involved. We consider these two questions in the context of a conversational intelligent tutoring system, CIRCSIM-Tutor. We first analyze the range of decisions that the system needs to make and define the information needed to support these decisions. We then describe four distinct models that provide different aspects of this information, taking into consideration the nature of the domain and the constraints provided by the tutoring system. At the end of the paper we briefly discuss our experiments with enhancing the student model in CIRCSIM-Tutor and some general problems regarding building and evaluating different student models.

Key Words: *Student Model, Intelligent Tutoring System*

1. Introduction

The content of student models in Intelligent Tutoring Systems (ITS) varies widely. Some student models are built for recognizing student plans or solution paths [2], some are built for evaluating student performance or problem solving skills [5], and some are created for describing constraints that the student has violated [9] [11]. But there is one question that must be answered whenever we begin to build a new student model: what aspects of the student should we model in a specific intelligent tutoring system?

Many researchers argue that the main purpose of a student model in the context of intelligent tutoring is to guide pedagogical decision-making. There is little need for a description of the student's knowledge unless there is some way for the system to make use of, or react to, that description [10] [13]. This functional view of the student model prompts the following question: if the tutoring system has several modules or layers for making different decisions, should the student model be similarly structured into different models?

In this paper we address what to model and how to divide the student model into components in the context of a dialogue-based intelligent tutoring system, CIRCSIM-Tutor. We analyze the variety of decisions that the system needs to make and extract the necessary information needed to support these decisions. We then describe four distinct models that provide different aspects of this information, taking into consideration the nature of the domain and the constraints imposed by the tutoring system. At the end of the paper, we discuss some general issues related to these two questions, illustrated with examples from two CIRCSIM-Tutor experiments.

2. About CIRCSIM-Tutor

We are building a new version of our intelligent tutoring system, CIRCSIM-Tutor, which helps medical students understand the negative feedback system that controls blood pressure. CIRCSIM-Tutor tutors by having students solve problems. Figure 1 shows the main interface of CIRCSIM-Tutor. The system presents the student with a description of a physiological change in a window located on the upper-right corner. For example, the patient might hemorrhage and lose a liter of blood. The tutor then asks the student to predict the effect of that change on seven important physiological parameters. These changes are entered in the *prediction*

Proceedings of the 11th IEEE International Conference on Tools with Artificial Intelligence. Chicago, IL, 1999, pp. 13-18

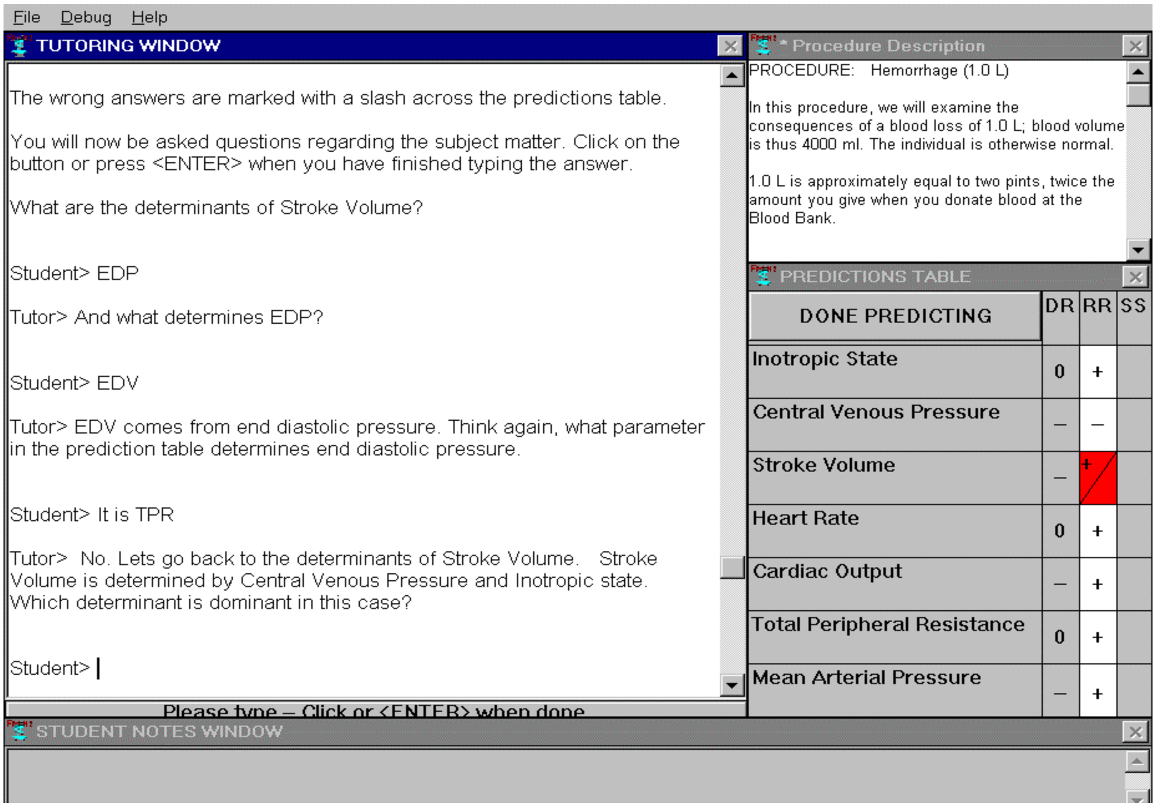

Figure 1. The main interface of CIRCSIM-Tutor

table located on the right of the main interface. The tutor conducts a dialogue with the student to correct the errors in the student's predictions. Depending on the tutoring protocol, this tutorial dialogue can occur after each erroneous prediction or after a larger hierarchical grouping such as a column or the entire table.

There are several facts about CIRCSIM-Tutor that influence the design of the student model:

- The goal of the tutoring system is to help students learn qualitative reasoning in a causal domain. In this domain, the causal relationships between parameters can be described at several levels.
- The students use a prediction table to solve the problem [8].
- The tutoring dialogue is plan-based and the tutor takes the lead most of the time.
- The tutoring dialogue is handled as free text input and output.

These features influence the information needed for tutoring decisions and constrain the possible information that the tutor can collect to build a student model.

3. Dividing the student model into components

3.1 Decisions in CIRCSIM-Tutor

We need to analyze what kinds of decisions have to be made in CIRCSIM-Tutor and what kind of information is needed to support these decisions. At the highest level of tutorial decision-making, CIRCSIM-Tutor has to make decisions such as choosing an appropriate problem for the student. At the same time, at the lowest level, as a dialogue-based ITS with free-text input and output, CIRCSIM-Tutor has to make decisions as detailed as how to choose an acknowledgment or a discourse marker. In a fully developed version of CIRCSIM-Tutor several modules will need to make different types of decisions: the curriculum planner, the tutorial planner, the turn planner, the discourse planner, and the surface generator. Each of these modules needs to communicate with the student model in order to make appropriate decisions. For some decisions, information about the student's overall performance is needed; while for other decisions, such as in discourse planning and surface generation, the most relevant information is the most recent statements of both student and tutor.

In the following sections we provide detailed analyses of some of the information needed by different modules in CIRCSIM-Tutor.

3.1.1 Adjusting the curriculum. To decide on the right difficulty level for the next problem, we may need the overall performance of the student [1]. But, to determine the description of the problem presented to the student, we may also need to use detailed knowledge of how much the student knows about specific issues in physiology.

3.1.2 Switching tutoring protocols. We have three different protocols; each of them has different constraints about how the tutor assists the student in solving the problem [6]. The protocol determines the number of related variables predicted by the student before the system starts to tutor the mistakes. Thus the decision to switch protocols may be governed by the student's performance on the causal relationships between variables.

3.1.3 Planning the tutoring dialogue. The detailed tutoring dialogue in CIRCSIM-Tutor is plan-based and controlled by the tutor most of the time, but new goals can emerge in an opportunistic fashion in response to unexpected student input. In CIRCSIM-Tutor, the student learns how to solve a problem in a causal domain. In this domain the causal relations between two parameters can be described at various levels of detail.

The tutor's decision about which way to teach is mostly determined by how the student has replied and in which order. If the student invokes an intermediate variable, we need to record where along the causal link that variable stands, so the tutor can continue from that point. Sometimes the tutor will invoke an intermediate variable by way of a hint. These intermediate causal relationships are not usually taught, but as soon as they are mentioned, it becomes important to teach them correctly. Thus the intermediate variables the student mentions, as well as their order, must be recorded in the student model.

3.2 Four components of the student model

From the above analysis, we can see that different types and levels of student information are needed for different kinds of decisions. Some of the information is stored numerically and some symbolically, while some must be kept as detailed history records. After analyzing the information needed by the system, based on the nature of the domain and the structure of tutorial decision-making, we designed four student models that offer different information. Below are the student models for CIRCSIM-Tutor.

3.2.1. Performance model. The performance model computes an assessment of student competence, based on performance. To determine the levels of this evaluative model, we considered mainly the tutoring structure of CIRCSIM-Tutor. In this system, the overall organization of a tutoring session is determined by the following structure:

- The student or the curriculum planner chooses problems for the student. Using medical terminology, each of the problems is called a *procedure*.
- The student then solves each procedure in three stages.
- In each stage, the student makes predictions about seven important physiological parameters and enters them in the prediction table.
- The tutor corrects incorrect predictions one by one and tutors the relationships between the concepts.

From this overall tutoring structure, it is natural to divide the evaluation of the student into four levels:

- Global assessment, an overall measurement of the student's ability
- Procedure-level assessment, a measurement for each problem the student is asked to solve
- Stage assessment, a measurement for each of the three physiological stages in a problem
- Local assessment, a measurement for each variable that has been tutored

The local assessment is updated after each tutoring interaction, and the other assessments are calculated from the local assessment. The assessment model is currently based on a set of simple heuristics. The local assessment depends on fifteen different patterns found in the student's answers. For example, if the student gets one parameter correct but another one wrong in his or her reply, then increase the local assessment by 0.1. A statistical model or an inference model could be used to propagate the local assessment to each of the upper level assessments.

3.2.2. Student reply history. A student reply history is attached to each concept CIRCSIM-Tutor teaches. To decide what to record in this model, we mainly consider the nature of the domain. In our domain, a causal relation between two parameters can be described at various levels of detail. The history model records what part of the causal chain has been covered as well as a history of student answers about that chain. For each student answer we also record its classification. Student answers are classified into eight categories to help the tutor decide on a response strategy [14]. For example, a near miss answer is one that is pedagogically useful but not the desired answer.

3.2.3. Student solution record. This model records how many errors the student made while solving the problem, along with a detailed record of each error. In CIRCSIM-Tutor, we try to help the student to solve a problem in a logical order. So, it is important to compare the order of the student's predictions to any appropriate logical orders (which are not unique) and to analyze any inappropriate sequences on the part of the student. This model is similar to the idea of the constraint-based student model [11]. But we also group all similar errors and analyze all possible misconceptions.

This model is necessary because the student's solution path cannot be inferred from his or her predictions in the prediction table. It roughly reflects the student's ability to solve a problem logically. If we provide a way to force students to draw their solutions, then we may need to adopt another indicator of students' problem solving skill.

3.2.4. Tutoring history model. This model includes both the plan history and the discourse history. It does not contain direct knowledge of the student; rather, it is a record of what the system has done. We consider it as an extended part of the student model because it is important for tutorial decision-making [3] and it also implicitly reflects how much the student has done and how well.

The tutoring history is mostly constrained by the planning mechanism. Freedman [3] modeled CIRCSIM-Tutor's tutorial plan as a hierarchical structure of tutoring goals implemented as schemata. The tutoring hierarchy includes three levels: tutoring strategies, topics within the strategy, and text generation primitives (*inform* or *elicit*) for each topic. So for each procedure we propose to record the tutoring history according to this hierarchy.

The four models are stored separately for decision making, evaluation and possible comparison. The performance model is stored as a set of numbers, while the other models are stored as lists.

4. Making consistent decisions

When there are several student models available, each model may suggest different decisions. So, how to make consistent decisions is very important. For example, in CIRCSIM-Tutor, the overall performance of the student may indicate that the student is struggling and the tutor should give more direct information to prevent the student from becoming too lost; on the other hand, the recent student answer history may indicate that the student has some understanding of the current topic and so the tutor should encourage the student to do as much as possible by him or herself. Given this conflict, it is important to make consistent decisions. In CIRCSIM-Tutor, we have simple rules to decide which model is more important. For example, if the recent history indicates positive evidence that the student is doing well on a topic, then the tutor will always try to encourage the student to do more, even when the overall performance of the student is poor. This rule expresses our human tutors' belief that the tutor should encourage the student to do as much as possible. So, the student will always get a second chance after an error if he or she evinces any understanding of the topic at all. The study in [7] also indicated that human tutors let the students do most of the work in overcoming impasses, while at the same time providing as much assistance as necessary.

5. Using multiple student models to support tutoring decisions

In this section we describe how we use different student models to support decisions in CIRCSIM-Tutor. Specifically, we will discuss how to use different student models in the planning of tutorial dialogue designed to remediate erroneous student predictions.

From the student's errors and the tutoring history the planner will choose the high level tutoring methods. The next step is to plan detailed tutoring dialogue by using the answer category, the student's performance, and the student's reply history.

From the decision-making analysis, we have seen that the tutor has a variety of ways to tutor each causal relationship and the student's reply is a key to decide which way to teach. In Dialogue 1, the tutor is trying to tutor the causal link between Stroke Volume (SV) and Central Venous Pressure (CVP).

Dialogue 1:

T: What is the determinant of Stroke Volume?
S: It is EDP [end diastolic pressure]. *(near miss)*
T: And what parameter in the prediction table determines EDP?
S: EDV [end diastolic volume]. *(near miss in the wrong direction)*
T: EDV comes from EDP. Again, what determines EDP?

The tutor begins by asking the student a question. The student may give different answers. According to our classification scheme, the student model will classify the answer into one of eight categories. The tutorial planner can then decide what to do next. If the student's answer is totally incorrect, then the planner may use the student's performance to help decide further between giving the correct answer or giving a hint. In this example the student model classifies the student's answer as a near miss and puts it into the student reply history along with the detailed content of the answer. Based on this category information, the planner then decides to ask a question based on this near miss in order to help the student find the expected answer. Now the student can continue. Again, the student model will classify the student's answer and put it into the record and the planner will take action based on the classification and the reply history. The student's next answer returns to a link that has already been tutored, so the tutor acknowledges the relationship and asks a follow up question based on the last correct link instead of the student's latest response.

6. Experiments with CIRCSIM-Tutor

Two experiments with CIRCSIM-Tutor have been conducted with first-year students from Rush Medical College. In April 1998, 22 students used an earlier version of CIRCSIM-Tutor, developed in Lisp by Woo and others [12]. In that version the student model is a simple overlay model of the prediction table. In November 1998, 50 students used a new version of CIRCSIM-Tutor, which includes a new input understander, an updated tutorial planner, and a new student model. The new student model includes prototypes of the four models described earlier: the student performance evaluation model, the tutoring history, the student reply history, and the student solution record. A log file was produced for each student. Comparing the log files from these two experiments, we can see that the new student model improves tutoring quality even though the four implemented models are very simple. From studying the log files, we see that the new student models were important in the following ways:

- The evaluation of the student's performance helped decide when to give hints and when to give answers when the student cannot answer a question.
- The student reply history allows the tutor to end a dialogue and return to the original plan when the student could not continue along a causal link.
- The category of student answer, another aspect of the student reply history, is effective in helping to decide on different retry strategies. It recognizes near misses and other categories of answers that were once treated as totally incorrect answers.
- The tutoring history prevented the tutor from giving the same hint repeatedly.

For example, in Dialogue 2, which is extracted from the April experiment, the tutor responds to an unexpected answer by telling the student he or she is wrong, even though strictly speaking he or she is correct, since the student model does not have enough categories of student answers.

Dialogue 2:

T: What is the determinant of Stroke Volume?
S: It is EDP.
T: No. Stroke Volume is not determined by EDP. Stroke Volume is determined by Central Venous Pressure and Isotropic State.

On the other hand, in Dialogue 3, which is extracted from the November experiment, the tutor can generate an appropriate hint according to the detailed categories of student answers.

Dialogue 3:

T: What is the determinant of Stroke Volume?
S: It is EDP. (near miss answer)
T: And what parameter in the prediction table determines EDP?

But the log files of the new version also indicate some problems that we need to consider. For example, we need a long-term tutoring history to prevent picking the same tutoring method repeatedly. We also need more discourse history to prevent textual incoherence when topics are changed abruptly [3] [15].

7. Discussion

Determining what information to include in the student model is a system-dependent task. There are several constraints we have to consider. The first constraint is the nature of the domain: Is it a quantitative domain or a qualitative domain? Is the focus on causal reasoning as in our system or on pure facts? Another constraint is the structure of a tutorial session. Basically, we need to consider how the system interacts with the student: how the system presents the problem to the student, how much the system can observe about the student's solution, and whether the system uses natural language or not. A third constraint is the tutorial decisions that the system needs to make. Does the system need to plan the curriculum, to switch between tutoring protocols, to plan tutoring dialogue, or just give simple feedback without multi-step plans.

Another important issue is how to evaluate the student model. Since we have different student models, we may adopt different evaluation methods for them. We may even adopt different methods to evaluate the history models and the performance models.

8. Conclusions and future work

In this paper, we discussed how to build a student model composed of different sub-models by considering the actual demands of different decision-making components of the system. We discussed how to determine the knowledge required in each model by considering three types of constraints imposed by the system: the nature of the domain, the structure of the tutoring session, and the tutorial decisions that the system needs to make. We compared the quality of tutorial dialogues based on two versions of the student model in CIRCSIM-Tutor: one with only a simple overlay and one with prototypes of the four models described in this paper.

We are currently developing a new version of CIRCSIM-Tutor that will have a totally new planning engine [3] [4]. We plan to implement the student models fully in this new version and study further how to evaluate student models.

Acknowledgments

This work was supported by the Cognitive Science Program, Office of Naval Research under Grant No. N00014–94–1–0338, to Illinois Institute of Technology. The content does not reflect the position or policy of the government and no official endorsement should be inferred.

Joel A. Michael and Allen A. Rovick, professors in the Department of Physiology at Rush Medical College, are responsible for the pedagogical and domain knowledge in CIRCSIM-Tutor. They also served as expert tutors for the transcripts. We also want to thank Michael Glass, Reva Freedman, and other members of the CIRCSIM-Tutor project.

References

[1] Cho, B., Michael, J. A., Rovick, A. A., and Evens, M. W. (1999). A Curriculum Planning Model for an Intelligent Tutoring System. *Proceedings of the 12th Florida Artificial Intelligence Symposium (FLAIRS–99)*, Orlando, Florida, pp. 197–201.

[2] Conati, C., Gertner, A. S., VanLehn, K., and Druzdzel, M. J. (1997). On-Line Student Modeling for Coached Problem Solving using Bayesian Networks. *Proceedings of UM–97, Sixth International Conference on User Modeling,* Chia Laguna, Italy, pp. 231–242.

[3] Freedman, R. (1996). Interaction of Discourse Planning, Instructional Planning and Dialogue Management in an Interactive Tutoring System. Ph.D. diss., Dept. of EECS, Northwestern Univ., Evanston, Illinois.

[4] Freedman, R. and Evens, M. W. (1996). Generating and Revising Hierarchical Multi-Turn Text Plans in an ITS. *Intelligent Tutoring Systems: Third International Conference ITS '96*, Montreal, Quebec, pp. 632–640.

[5] Katz, S., Lesgold, A., Eggan, G., and Gordin, M. (1993). Modeling the Student in SHERLOCK II. *Journal of Artificial Intelligence and Education* (Special Issue on Student Modeling), 3(4): 495–518.

[6] Khuwaja, R. A., Rovick, A. A., Michael, J. A., and Evens, M. W. (1995). A Tale of Three Protocols: The Implications for Intelligent Tutoring Systems. Yfantis, E. A., Ed. *Intelligent Systems: Third Golden West International Conference*, Las Vegas, Nevada, 1994. Dordrecht: Kluwer Academic, pp 109–118.

[7] Merrill, D. C., Reiser, B. J., Ranney, M., and Trafton, J. G. (1992). Effective Tutoring Techniques: A Comparison of Human Tutors and Intelligent Tutoring Systems. *Journal of Learning Sciences* 2(3): 277–305.

[8] Michael, J. A., Rovick, A. A., Evens, M. W., Shim, L., Woo, C., and Kim, N. (1992). The Uses of Multiple Student Inputs in Modeling and Lesson Planning in CAI and ICAI Programs. Tomek, I., Ed. *Computer-Assisted Learning*: 4th International Conference, ICCAL '92, Wolfeville, Nova Scotia, pp. 441–452. Berlin: Springer-Verlag.

[9] Mitrovic, A. (1998). Experiences in Implementing Constraint-Based Modeling in SQL-Tutor. Goettl, B. P., Halff, H. M., Redfield, C. L., and Shute, V. J. Eds., *Intelligent Tutoring Systems*. 4th International Conference on Intelligent Tutoring System (ITS–98), San Antonio, Texas, pp. 414–423. Berlin: Springer-Verlag. (Lecture Notes in Computer Science, no. 1452.)

[10] Ohlsson, S. (1986). Some Principles of Intelligent Tutoring. *Instructional Science* 14: 293–326.

[11] Ohlsson, S. (1992). Constraint-Based Student Modelling, *Artificial Intelligence in Education*, 3(4): 429–447.

[12] Woo, C. (1991). Instructional Planning in an Intelligent Tutoring System: Combining Global Lesson Plans with Local Discourse Control. Ph.D. diss., Dept. of CSAM, Illinois Institute of Technology, Chicago, Illinois.

[13] Woolf, B. and Murray, T. (1994). Using Machine Learning to Advise a Student Model. Greer, J. E. and McCalla, G. I., Eds., *Student Modelling: the Key to Individualized Knowledge-Based Instruction*, Berlin: Springer-Verlag, pp. 127–146. (NATO ASI Series, series F: Computer and Systems Sciences, no. 125.)

[14] Zhou, Y., Freedman, R., Glass, M., Michael, J. A., Rovick, A. A., and Evens, M. W. (1999). What Should the Tutor Do When the Student Cannot Answer a Question? *Proceedings of the 12th Florida Artificial Intelligence Symposium (FLAIRS–99)*, Orlando, Florida, pp. 187–191.

[15] Zhou, Y., Freedman, R., Glass, M., Michael, J. A., Rovick, A. A., and Evens, M. W. (1999). Delivering Hints in a Dialogue-Based Intelligent Tutoring System. *Proceedings of the 16th National Conference on Artificial Intelligence (AAAI–99)*, Orlando, Florida, pp. 128–134.