Kong, S.C., Ogata, H., Arnseth, H.C., Chan, C.K.K., Hirashima, T., Klett, F., Lee, J.H.M., Liu, C.C., Looi, C.K., Milrad, M., Mitrovic, A., Nakabayashi, K., Wong, S.L., Yang, S.J.H. (eds.) (2009). Proceedings of the 17th International Conference on Computers in Education [CDROM]. Hong Kong: Asia-Pacific Society for Computers in Education.

# Towards a negotiable student model for constraint-based ITSs

**David THOMSON, Antonija MITROVIC**
*ICTG, University of Canterbury, Christchurch, New Zealand*
djt88@student.canterbury.ac.nz

**Abstract:** Much research has been done on open student models within adaptive educational systems. It has been shown that allowing the student to view their student model is useful in the learning process. Open student models help support meta-cognitive process, such as self-reflection. Negotiable student models take this a step further, and allow students to negotiate and potentially modify their model. A few negotiable student models have been implemented, but only in relatively simple systems, and not integrated into a complex ITS. As such, it is not clearly known if negotiable student models pose a significant advantage over the traditional open student models. This research implements a basic negotiable student model into a version of a complex and internationally deployed ITS. Subjective evaluation is performed, and shows promising results. Participants felt the negotiable student model was both useful for learning, and enjoyable to use. With a few improvements, this negotiable student model implementation could be used in a wide-scale objective analysis to help determine the usefulness of negotiable student models.

**Keywords:** Intelligent tutoring systems, negotiable student model, student modelling.

## Introduction

Intelligent Tutoring Systems (ITSs) are computer based tutors that aim to provide the same level of student specific help as a human tutor [4]. This is achieved through Artificial Intelligence, student modeling, and other methods [1]. In an ITS, the system tracks the student's actions, and builds a model of their knowledge. This model is then used to influence pedagogical decisions, such as which problem to suggest to the student next. This allows ITSs to adapt to students of differing ability levels: a below average student will get different recommendations, and different feedback than an above average student. ITSs aim to give feedback appropriate to students of all abilities, so a struggling student will be given substantial assistance while a competent student will be given less. Recommending questions based on the students ability means that struggling students will not get overwhelmed, and more successful students will not get bored. ITS can therefore cater to different learning styles, although some are be better (in respect to the amount of material covered correctly) than others [7].

At least four different levels of visibility for the student model exist: Hidden, Open, Editable, and Negotiable. Often in ITSs the student model is hidden from the student, and is used only by the system itself. It has however, been shown that allowing the student to view their model increases the student's learning [6]. Editable student models allow the student not only view their model, but to change it whenever they believe it does not represent their knowledge accurately. Negotiable student models allow a form of editing on the model, but the student must convince the system that their knowledge is correct before any changes are made. The purpose of opening up the student model is to get the student to be actively involved in their learning and self assessment. In performing these meta-cognitive processes, the student is likely to learn more from the ITS [6], as improved meta-cognitive

skills lead to an improvement in learning. Research has been done on how to best display an open student model [2, 10, 15], but few ITSs implement a negotiable student model.

## 1. Background

### *1.1 Open Student Models*

Student modeling can be defined as the process of gathering relevant information in order to infer the current cognitive state of the student, and to represent it so as to be accessible and useful to the pedagogical module.

Although computing a complete and correct student model is intractable [5], a useful model can still be implemented effectively. This can be achieved when it is realised that the usefulness of the model is more important than its completeness. When constructing a student model the ITS should: avoid guessing, not bother to diagnose what it can not treat, and empathise with the student [5]. This results in dynamic student models that are as accurate as needed, and can be used by the system (specifically the pedagogical module) to make pedagogical decisions.

Open, inspectable, or viewable student models extend the purpose of a student model from a source of information for the system to a source of information for the student (and the system) [12]. An open student model reflects to students feedback on their progress and overall performance in the system. Usually, the student model is broken up into categories, or concepts. Student performance and progress is shown for each concept. This allows the student to see their strengths and weaknesses within the domain on a finer level. The student will be able to see where their strengths and weaknesses are, and therefore which material to focus on. As well as passively suggesting learning material to the student, an open student model aims to promote self-reflection and assessment through inspection of the model.

### *1.2 Related Work*

A number of negotiable student models have been implemented, including: CALMsystem [16], StyLE-OLM [17], and Mr. Collins [9]. These systems all have a similar approach to negotiable student modelling. The student model is split into two components, which may or may not become integrated. One component is maintained by the student, and the other by the system. This approach effectively involves two student models, one constructed by the student's own self-assessment, and the other by the systems calculations. If corresponding sections of the models differ by more than some predetermined amount, the student and system will engage in a dialog to try to agree on a common value. These dialogs might include explanations and justifications, of both the students and the system's beliefs.

### *1.3 EER-Tutor*

EER-Tutor is a web-enabled ITS that teaches Enhanced Entity-Relationship (EER) modelling. EER-Tutor is based on Contraint-Based Modelling, and is used at the University of Canterbury and through a web portal at DatabasePlace (http://www.aw-bc.com/databaseplace). Enhanced Entity Relationship modeling is an ill-defined, open-ended task. This means the start and end states, as well as operators are difficult to define. The problem solving algorithms are underspecified, and most problems

will have more than one correct solution. Regardless, EER-Tutor has been shown to be effective when combined with traditional lectures [11].

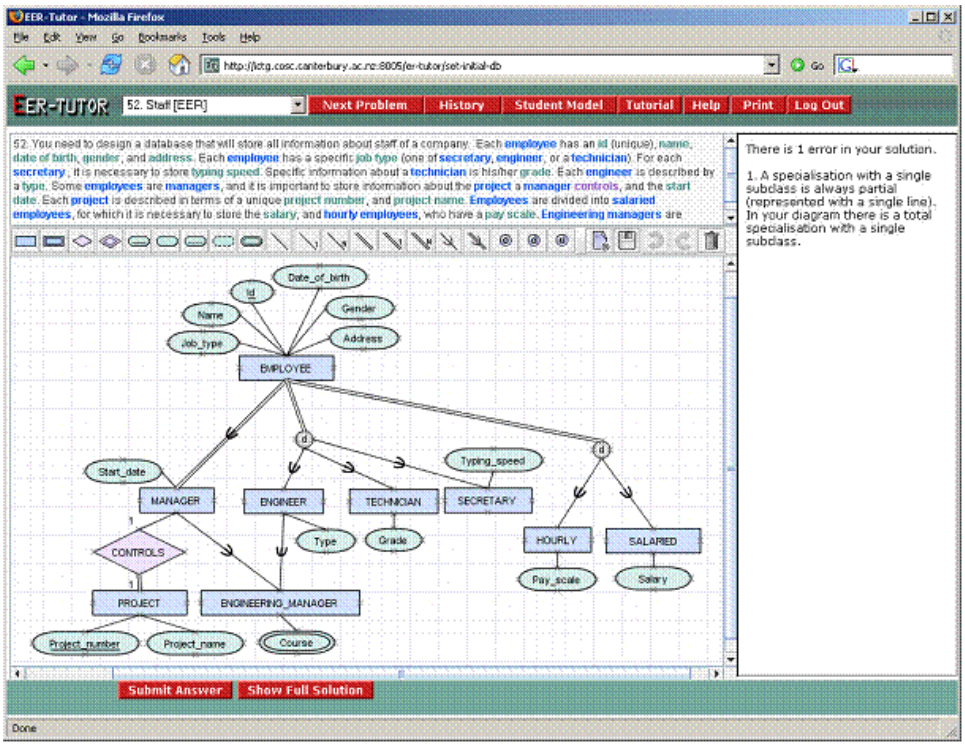


Figure 1: The EER-Tutor interface

Figure 1 shows the EER-Tutor interface. The main area of EER-Tutor is a place for the student to draw EER diagrams. Tool buttons are provided for the different components of an EER diagram, and the question text is always shown. These two features aim to help reduce the cognitive load on students. When the student wishes, they can submit their diagram. If there are any errors in their solution, feedback will be displayed on the right side of the window. The student can use this feedback to help correct their solution, before re-submitting. There are buttons for system actions such as Next Problem, a Tutorial, Help, and Logout. There is also a button for the student to view their student model, shown in Figure 2.

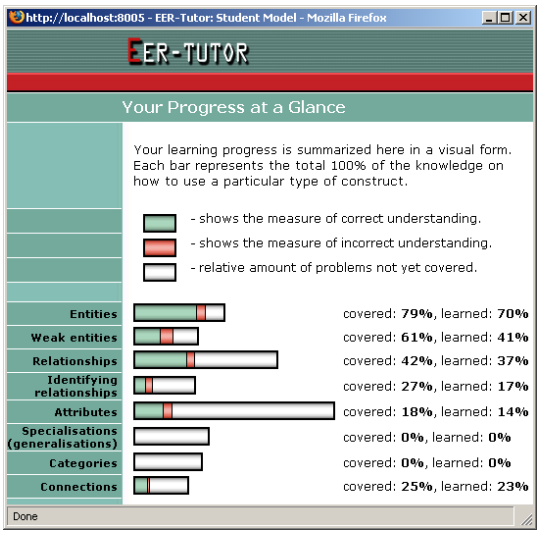


Figure 2: The open student model in EER-Tutor

The domain (Enhanced Entity Relationship modelling) is broken down into eight concepts. The student is able to see which specific parts of the domain they have covered, and to what level of proficiency. The horizontal size of the bar indicates how much material there is on that concept in the tutor. This bar is divided into three distinct sections. The first section (a predominant green colour) represents correct knowledge; the second section (bold red) represents incorrect knowledge. The amount of material not yet covered by the student is represented by the third (white) section. As the student progresses through the tutor, the total material covered (correct understanding plus incorrect understanding) will increase. Hopefully, but not necessarily, the amount of incorrect understanding will decrease, until all the material is covered correctly.

Where previous research [9, 16] has implemented a negotiable student model in a simple computer-based learning system, this project uses EER-Tutor, a complete ITS. Instead of multi-choice questions EER-Tutor presents a problem solving environment that the student must use to solve questions. EER-Tutor has a complex solution evaluator and gives dynamic feedback based on the student's actions within the system. This allows for evaluation of a negotiable student model in a complex ITS that is being used in a university course, and internationally over the Internet.

## 2 Enhancing EER-Tutor

A negotiable student model allows the student to edit their model, but there needs to be a form of control on this editing. If the student could arbitrarily change their model, it would defeat the purpose of the model which is to reflect the current knowledge of the student. One way to implement this control is to force the student to first convince the system of their knowledge, before any modifications are made. If the student does not agree with part of their model, they can start a dialog with the system. If the student can convince the system their knowledge is higher than their model suggests, the system will modify the model appropriately.

The negotiable student model in EER-Tutor has been designed as an additional, separate component. The existing student model is still used, with the negotiable student model effectively acting as a layer above the conventional model. When the student views their model, they see a combination of the two models. Changes made to the negotiable student model have no effect on the underlying student model, which can no longer be seen by the student. This design was chosen to minimise the effect of adding a negotiable student model to EER-Tutor.

The interface for the enchanced version of EER-Tutor is shown in Figure 3. The main change is that the student model is now always displayed to the student. This was a deliberate change, and should prompt the student to think more about their learning, thus increasing their meta-cognitive processes.

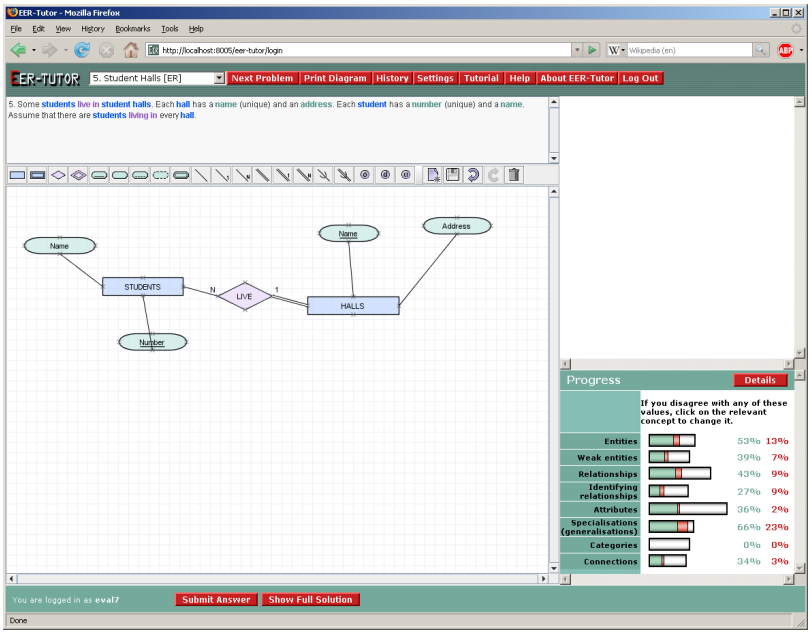


Figure 3: The Enhanced EER-Tutor interface

If at any stage the student feels any part of their student model does not reflect their actual knowledge, the student may enter a dialog with the system. This is done by clicking on one of the concepts from the student model. The system asks the student a question relating to that concept, which, if they answer correctly, will increase their correct understanding for that concept. If the student fails to answer the question correctly in the specified number of attempts, the system will decrease the correct understanding for that particular concept.

When using the negotiable student model, it is only possible to change the correct and incorrect components, but not material covered. At best, one can eliminate all the incorrect (red) knowledge. This means that to cover more material it is still necessary to attempt domain problems. This helps to ensure that the negotiable student model does not become the student's focus; they still need to work on domain problems to progress through the tutor. The system currently supports two types of questions: multi-choice and short answer. Figure 4 shows an example of a short answer question.

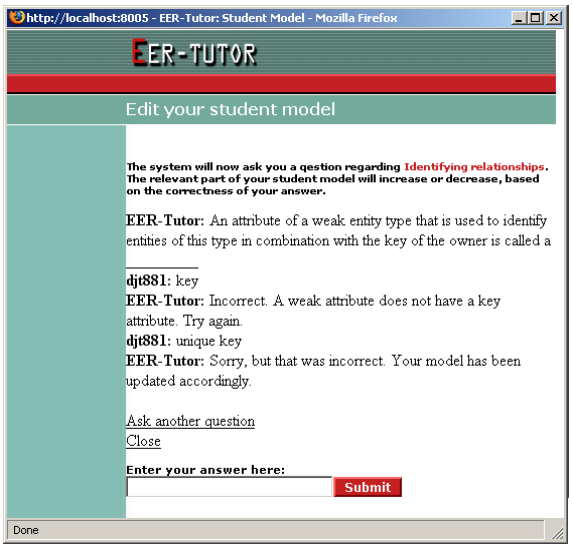


Figure 4: A short answer question

Every question (short answer and multi-choice) has six components: a question number, relevant concept, the question text, the correct answer(s), incorrect answer and feedback pairs, and a maximum number of attempts, as shown in Figure 5. Figure 6 shows the actual representation for question number 3.

```
(question-number
 relevant-concept
 (question-text (option1 option2))
 (correct-answer1 correct-answer2 …)
 ((incorrect-answer1 feedback1)
  (incorrect-answer2 feedback2) …)
 max-number-of-attempts)
```

Figure 5: General question structure

```
(3
 "identifying relationships"
 ("An attribute of a weak entity type that is used to identify entities of
this type in combination with the key of the owner is called a _________"
nil)
 ("partial key")
 (("key" "Incorrect. A weak attribute does not have a key attribute.")
  ("unique" "Incorrect. A weak attribute does not have any unique
attributes.")
  ("primary key" "Incorrect. A weak attribute does not have a key
attribute."))
 2)
```

Figure 6: Question number 3

Behind the scenes, both multiple-choice and short answer questions are dealt with in the same way. For multiple-choice questions the interface generates the textual answer corresponding to the item the student selected. This means all questions and answers are processed in the same way, keeping the design and code consistent. This has been done to make the possible future addition of a natural language parser as easy as possible. Each question can have feedback, specified by the author, for specific incorrect answers. If the student's answer matches one of these incorrect answers, the corresponding feedback will be displayed (as in Figure 4).

## 3 Evaluation

To conclusively evaluate this system a thorough objective evaluation should be performed. However, due to the timing of this project, it was not possible to run such an evaluation. A subjective survey was conducted, seeking opinions from students concerning the negotiable student model. Eleven participants were involved; three experts (people involved in the development of EER-Tutor), and eight volunteers. Participants were asked to use the system for an undetermined period of time, until they had a good feel for the system. They then completed a questionnaire, which consisted of ranking the system on five aspects, and some open-ended questions which aimed to give the opportunity for participants to voice their opinions on the system. All participants (both experts and volunteers) were postgraduate Computer Science students.

Table 1: Survey results

| Question | All (n:11) | Experts (n:3) | Others (n:8) |
|---|---|---|---|
| Did you enjoy learning with EER-Tutor? | 4.0 (1.0) | 4.7 (0.4) | 3.8 (1.1) |
| Did you find the NSM interface easy? | 4.5 (0.5) | 4.3 (0.4) | 4.5 (0.5) |
| Did the NSM help you to learn? | 3.9 (0.9) | 3.7 (0.4) | 3.9 (0.5) |
| Was the behaviour of the NSM logical? | 3.9 (0.9) | 4.3 (0.9) | 3.8 (0.5) |
| Did you find the NSM distracting? | 1.5 (0.7) | 2.0 (0.7) | 1.1 (0.7) |

Table 1 shows the average ranking of aspects of EER-Tutor with the negotiable student model (NSM). The standard deviation is shown in parentheses. For each question, participants were asked to select a value on a scale of one to five, with one representing *not at all*, and five representing *very much*. All of these results are positive. Students found the negotiable student model easy to use, helpful, reasonably logical, and not distracting. The expert ratings were not consistently different to the non-expert ratings, with some values being higher and some lower than the overall average.

On average participants used the system for 42 minutes each, and answered a total of 115 questions, at an average of 10.5 questions each. The time spent by each participant ranged from 5 minutes up to 157 minutes. The three experts spent 35, 12, and 157 minutes using the system, and answered 10, 4, and 6 questions respectively. In this context, questions are related to the negotiable student model, not domain questions.

## 4 Conclusion

This project designed and implemented a negotiable student model in EER-Tutor. Much research has been done on open student models [8, 9, 13, 14], but no negotiable student model has yet been implemented and evaluated in a large scale ITS. This project sets the stage for this evaluation to take place. A negotiable student model has been implemented, and evaluated subjectively with a user questionnaire. Next, it should be thoroughly, objectively analysed, to determine if it is beneficial to the learning process. If it is shown that a negotiable student model does help students learn, this research could be used as a basis for implementing a negotiable student model in other ITSs.

The negotiable student model implemented in this project was designed to be simple enough to be implemented in a short time frame. As a result, it is by no means a feature-complete negotiable student model. Many enhancements and new features could be added.

Although no objective data has been collected, subjective results have been very positive. Almost all participants felt the negotiable student model would help them learn, and some noted it was a nice break from problem solving, and encouraged them to correct their knowledge. This enjoyment and added motivation in itself is important in any learning situation. Negotiable student models are welcomed by users, and should be considered for any ITS.

## References

[1] Beck, J., Stern, M., Haugsjaa, E. Applications of AI in Education. Crossroads (special issue on AI), vol. 3(1), pp. 11-15, 1996.
[2] Bull, S., Cooke, N., Mabbott, A. Visual Attention in Open Learner Model Presentations: An Eye-Tracking Investigation. In C. Conati, K. McCoy & G. Paliouras (Eds.): User Modeling 2007: 11th International Conference, Springer-Verlag, Berlin Heidelberg, pp. 187-196, 2007.
[3] Bloom, B.S. The 2 Sigma Problem: The Search for Methods of Group Instruction as Effective as One-to-One Tutoring. Educational Researcher, vol. 13, 3-16, 1984.
[4] Mitrovic, A., Martin, B., Suraweera, P. Intelligent tutors for all: Constraint-based modeling methodology, systems and authoring. IEEE Intelligent Systems, special issue on Intelligent Educational Systems, vol. 22, no. 4, 38-45, July/August 2007.
[5] Self, J. A. Bypassing the intractable problem of student modeling. In C. Frasson & G. Gauthier (Eds.): Intelligent tutoring systems: At the crossroads of artificial intelligence and education, pp. 107-123. Norwood, NJ: Ablex, 1990.
[6] Mitrovic, A., Martin, B. Evaluating the Effect of Open Student Models on Self-Assessment. Int. J. Artificial Intelligence in Education, vol. 17, 121-144, 2007.
[7] Mathews, M., Mitrovic, A., Thomson, D. Analyzing high-level help seeking behaviour in ITSs. In W. Nejdl et al. (Eds.): AH 2008, LNCS 5149, pp. 312-315, 2008.
[8] Bull, S. See Yourself Write: A Simple Model to Make Students Think. In A. Jameson, C. Paris and C. Tasso, (Eds.): User-Modeling: Proceedings of the Sixth International Conference (UM97). New-York: Springer, pp. 315-326, 1997.
[9] Bull, S., Pain, H. Did I say what I think I said, and do you agree with me?: Inspecting and Questioning the Student Model. In Y. Greer (Ed.): AIED Proceedings, pp. 501-508, 1995.
[10] Bauer, M., Gmytrasiewicz, P.J., Vassileva, J. Supporting Negotiated Assessment Using Open Student Models. UM2001, LNAI 2109, 295-297, 2001.
[11] Suraweera, P., Mitrovic, A. An Intelligent Tutoring System for Entity Relationship Modeling. Int. J. Artificial Intelligence in Education, vol. 14, no. 3-4, 375-417, 2004.
[12] Bull, S., Dimitrova, V., McCalla, G. Preface for Special Issue (Part 1) Open Learner Models: Research Questions. Int. J. Artificial Intelligence in Education, vol. 17, 83-87, 2007.
[13] Bull, S., Kay, J. Student Models that Invite the Learner In: The SMILI Open Learner Modelling Framework. Int. J. Artificial Intelligence in Education, vol. 17, 89-120, 2007.
[14] Lazarinis, F., Retalis, S. Analyze Me: Open Learner Model in an Adaptive Web Testing System. Int. J. Artificial Intelligence in Education, vol. 17, 255-271, 2007.
[15] Van Labeke, N., Brna, P., Morales, R. Opening up the Interpretation Process in an Open Learner Model. Int. J. Artificial Intelligence in Education, vol. 17, 305-338, 2007.
[16] Kerly, A. & Bull, S. Children's Interactions with Inspectable and Negotiated Learner Models. In B.P. Woolf, E. Aimeur, R. Nkambou & S. Lajoie (Eds.): Intelligent Tutoring Systems: 9th International Conference, Springer-Verlag, Berlin Heidelberg, pp. 132-141, 2008.
[17] Dimitrova, V. STyLE-OLM: Interactive Open Learner Modelling. Int. J. Artificial Intelligence in Education, vol. 13, 35-78, 2003.
[18] Parker, L. Little wonders. Australian Educator, Spring 2008, 18-20, 2008.
[19] Bennet, F. Computers as tutors: solving the crisis in education. Faben, 1999.