Research and Practice in Technology Enhanced Learning
Vol. 5, No. 1 (2010) 19–33

DOI: 10.1142/S1793206810000797

PRELIMINARY EVALUATION OF A NEGOTIABLE STUDENT MODEL IN A CONSTRAINT-BASED ITS

DAVID THOMSON* and ANTONIJA MITROVIC†
Intelligent Computer Tutoring Group
University of Canterbury, Christchurch, New Zealand
**djt88@uclive.ac.nz*
†*tanja.mitrovic@canterbury.ac.nz*

Previous research on adaptive educational systems has shown that allowing the student to view their student model is useful in the learning process. Open student models help support meta-cognitive processes, such as self-assessment and reflection, and at the same time increase the student's trust in the system. Negotiable student models take this a step further, and allow students to negotiate and potentially modify their model. Very few negotiable student models have been implemented, and only in relatively simple systems, not integrated into a complex Intelligent Tutoring System (ITS). Therefore, it is not clearly known whether negotiable student models pose a significant advantage over simpler 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; constraint-based tutors; negotiable student model; student modelling.

1. Introduction

Intelligent Tutoring Systems (ITS) are computer-based educational systems that aim to provide the same level of student specific help as a human tutor (Mitrovic *et al.*, 2007). This is achieved through Artificial Intelligence, student modeling, and other methods (Beck *et al.*, 1996). An ITS tracks the student's actions, and builds a model of their knowledge. This process is known as student modeling. The student model is used to influence pedagogical decisions, such as which problem to suggest to the student next. Student modeling allows ITSs to adapt to students of different 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 student's ability means that struggling students will not get overwhelmed, and more competent students will not get bored. ITSs can therefore cater to different learning styles, although some are better (in respect to the amount of material covered correctly) than others (Mathews *et al.*, 2008).

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; the student is not even aware of the existence of the student model. It has however been shown that allowing students to open their models increases the student's learning and improves their meta-cognitive skills (Mitrovic & Martin, 2007). Open student models are visual representations of the content of the student model, which the student can inspect, but cannot change. Editable student models, on the other hand, allow the student not only to view their model, but to change it whenever they believe it does not represent their knowledge accurately. The problem with this approach is that students often over- or under-estimate their knowledge, which results in inaccurate student models. In order to avoid this problem, negotiable student models engage the student in a negotiation process. If the student disagrees with the system's estimate of their knowledge, he/she will get additional questions from the system. Based on the student's answers, the system can then adjust the student model.

The purpose of opening up the student model is to encourage the student to be actively involved in their learning and self assessment. Giving the student access to their model allows the student to reflect on their knowledge. In performing these meta-cognitive processes, the student is likely to learn more from the ITS (Mitrovic & Martin, 2007), as improved meta-cognitive skills lead to an improvement in learning. Metacognition has been studied in several disciplines, such as Education, Psychology and Artificial Intelligence. It is generally accepted that metacognition includes the processes and activities involved with awareness of, reasoning and reflecting about, and controlling one's cognitive skills and processes. Metacognition therefore involves thinking about, inspecting and adjusting one's thinking, problem-solving approaches and learning habits, among others. Studies show that improved meta-cognitive skills can be taught (Bielaczyc *et al.*, 1993) and result in improved problem solving and better learning (Swanson, 1990). A lot of research has been done on how to best display an open student model (Bull *et al.*, 2007; Bauer *et al.*, 2001; Van Labeke *et al.*, 2007), but very few systems implement a negotiable student model.

We start by reviewing briefly the related work on open/negotiable student models. In Sec. 2 we present EER-Tutor, an ITS which is the context of our research. Section 3 presents a basic negotiable student model we developed and integrated into EER-Tutor. We performed the preliminary, subjective evaluation of the negotiable model, and present its results in Sec. 4. We end with a conclusion in Sec. 5.

2. Related Work

Student modeling can be defined as the process of gathering relevant information in order to infer the current cognitive state of the student, and representing it so as to be accessible and useful to the pedagogical module. Although generating a complete and correct student model is intractable (Self, 1990), a useful student model 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 (Self, 1990). 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) (Bull *et al.*, 2007). An open student model shows feedback to students on their progress and overall performance. 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, and therefore which material to focus on. As well as passively suggesting learning material to the student, an open student model aims to promote reflection and self-assessment through inspection of the model.

Several existing ITSs open their student models to students (Mitrovic & Martin, 2007; Aleven & Koedinger, 2000; Brusilovsky *et al.*, 1996) but none include negotiable student models. Research on negotiable student models has only been done in the context of educational systems where the primary student activity is question answering, or model building, rather than problem solving. We now present a few related projects.

2.1. *CALMsystem*

CALMsystem (Kerly & Bull, 2007) implements a negotiable student model, aimed at primary school students. The system asks multi-choice questions in order to assess the student on each topic. The student provides their own estimate of their knowledge of a topic, while the system also scores their knowledge based on the answers to questions. These two scores, on a scale of zero to one, are converted into "low", "moderate", "good", or "high". Students view these scores (represented by pictures), and can quickly see discrepancies between their beliefs and those of the system. Students can negotiate their model through a conversational agent, using a natural language interface. Beliefs are changed through discussion with the agent, and may be initiated either by the system or the student. Students must explicitly rank themselves after every question, and can then compare their rankings with the systems. If the student wishes, they can get justification from the system for the system's beliefs. To change the system's beliefs, the student must answer questions, and the system will modify its beliefs based on the responses.

An experimental evaluation was conducted with a tutor loaded with science questions, and 25 UK Primary school children, aged 10–11. The evaluation showed that both the negotiable student model version of CALMsystem, and a version with an inspectable student model helped to improve student's meta-cognitive processes. The study found that users of the negotiable version of CALMsystem reduced inaccuracies in their self-assessments significantly more than users of the version without negotiation support.

2.2. *STyLE-OLM*

STyLE-OLM (Dimitrova, 2003) integrates a negotiable student model into STyLE, an adaptive knowledge-based web learning environment aimed at assisting learners from Bulgaria, Romania and Ukraine in acquiring Finance terminology in English. STyLE-OLM features a student model jointly constructed by the system and the student, which contains the student's beliefs and misconceptions. Students are shown a graphical representation of their beliefs, as well as textual information. The student is involved in the construction of their model through dialogue with the system, where both the student and system can ask questions, state propositions, and challenge or justify claims. STyLE-OLM maintains just one student model, so the system and student must agree at some level. To achieve this, a complex process involving reasoners and an initial set of beliefs is used. A small evaluation of STyLE-OLM was conducted, which mainly focused on the behaviour of the system, and not the effectiveness of the student model on learning. In general, STyLE-OLM provided an adequate environment for inspecting and discussing the student model. However, some of the natural language dialogue confused or frustrated some of the users.

2.3. *Mr. Collins*

Mr. Collins (Bull & Pain, 1995) implements a collaboratively maintained, inspectable student model. In this system the student model is maintained by both the student and the system. Mr. Collins uses a simple student model, which contains two separate confidence measures. The first is provided by the student, and reflects the student's current belief of their knowledge. The second measure is calculated by the system based on the student's performance. These confidence measures take the form of a value from a four point scale (very sure/almost sure/unsure/very unsure). If the two measures differ by a significant amount (more than one value difference on the scale), the student enters a dialog with the system. Here the student can choose to change their own beliefs, or challenge the system's beliefs. When changing the student's own beliefs, the student can ask the system to justify its decision, which may involve showing the student's last five attempts on the relevant problem. When challenging the system's beliefs, the student may have to justify themselves, which consists of answering a question. The domain for Mr. Collins is object pronouns in European Portuguese for second language learners, and there

are twelve rules for pronoun placement in the system. This research investigated whether students would inspect their own student model, and whether they would challenge the contents of the model in cases where they disagreed. Results showed that students did in fact inspect and challenge their model.

3. EER-Tutor

EER-Tutor is a web-enabled Intelligent Tutoring System that teaches conceptual database design using the Enhanced Entity-Relationship (EER) model. EER-Tutor is a constraint-based tutor (Mitrovic *et al.*, 2007), and is used at the University of Canterbury and through a web portal at DatabasePlace (http://www.aw-bc.com/databaseplace). The system complements traditional lectures by providing individualized problem-solving opportunities to students. Conceptual database design is an ill-defined, open-ended task, as its start state (i.e. the database requirements) is often ambiguous and incomplete, the end state (i.e. the database schema) is defined in abstract terms, and there is no algorithm for turning the start state into the goal state. Additionally, most problems have more than one correct solution. EER-Tutor has been shown to significantly enhance students' learning in as little as two hours of interaction (Suraweera & Mitrovic, 2004).

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 working memory load, as students can remind themselves of the problem requirements and select appropriate components. 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. The system uses the domain model represented as a set of constraints in order to diagnose student solutions. 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. For more details about the system please see (Mitrovic *et al.*, 2007; Suraweera & Mitrovic, 2004).

The open student model (shown in Figure 2) shows the abstract view of the student's knowledge in the form of eight skill meters, representing the domain at a high level of granularity. 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,

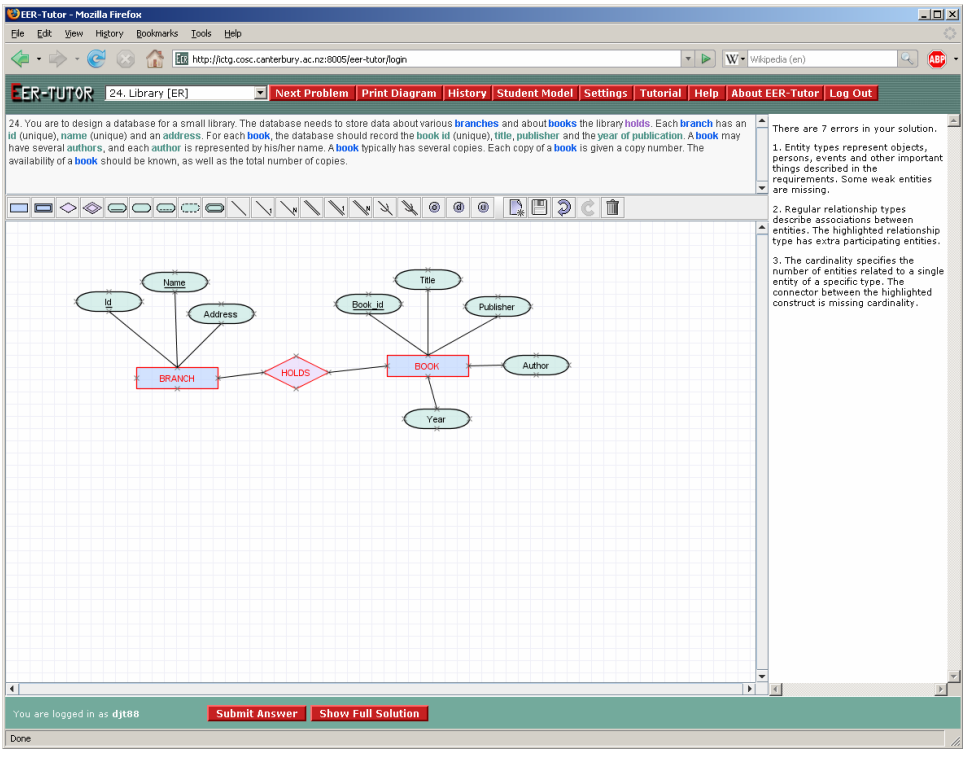

Figure 1. The EER-Tutor interface.

until all the material is covered correctly. Previous studies with EER-Tutor have shown that students understand the open student model and find them a useful additional tool for their learning (Mitrovic & Martin, 2007).

4. Designing a Negotiable Student Model for EER-Tutor

A negotiable student model allows the student to change their model, but there needs to be a form of control on these changes. 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 model will be modified.

Figure 3 shows an example of a possible dialog between the student and the system. The students states they believe the system's model of their knowledge is incorrect, so initiates a dialog with the system. The system asks the student a question regarding the *entities* concept. The student's first attempt ("*big words*") is

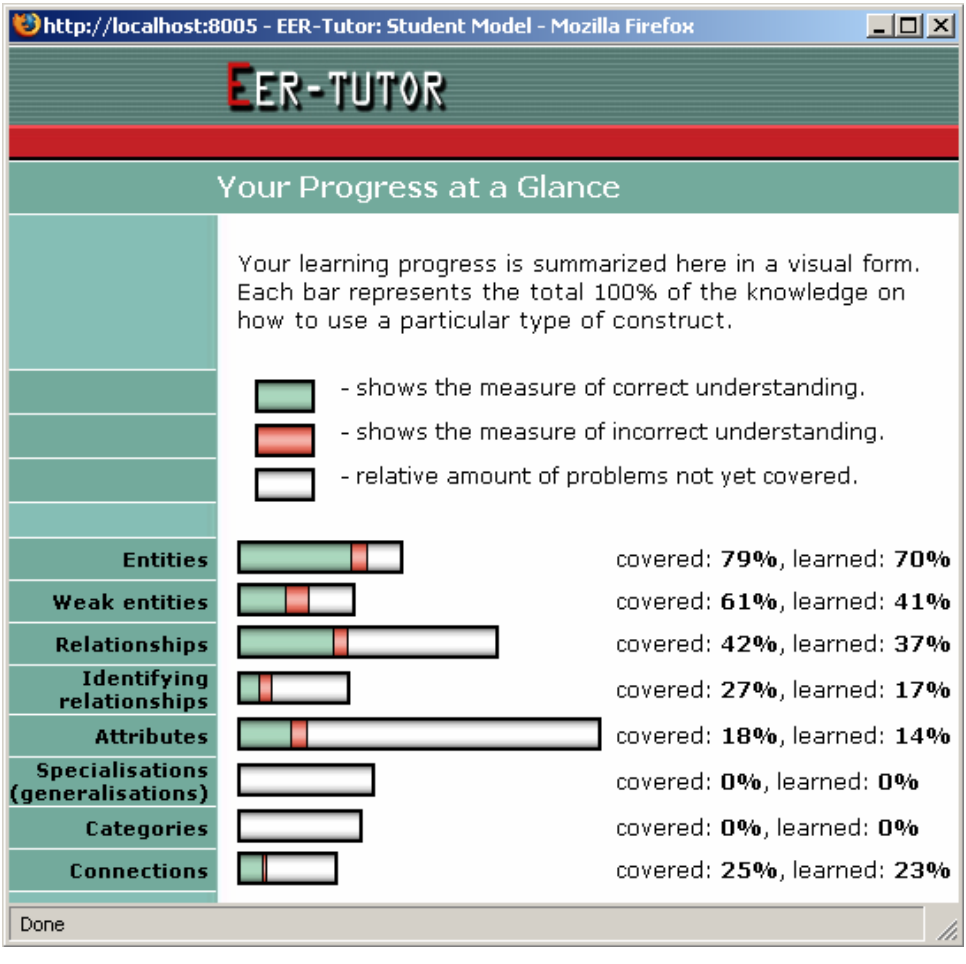

Figure 2. The open student model in EER-Tutor.

incorrect; the system gives them a hint and prompts them to answer again. On the second attempt the students answer is correct, at which stage the system updates the student model.

The negotiable student model in EER-Tutor has been designed as an additional, separate component. This is because the negotiable student model and the regular student model relate to different student activities. The regular student model is concerned with problem solving and records when the student violates and satisfies constraints. On the other hand, the negotiable student model records how well the student has answered questions when discussing their model. These questions are based on concepts rather than practical problem solving, so the information is kept in a different place. In the future it may be possible to reduce or remove the distinction between the two models, but further investigation is required.

```
Student: "I know more than you think I do about
entities."

System: "Ok, when drawing EER diagrams, which
sort of words in the problem text are likely to
model entities?"

Student: "big words"

System: "Not quite, the answer is a type of word
e.g. verb, adjective, noun etc. Try again."

Student: "nouns"
```

Figure 3. An example dialog.

When displaying the model to the student, the two sets of information are combined. For each concept in the student model, the value for the corresponding concept in the negotiable student model is added to the correct knowledge component of that concept from the regular student model. More detail on how the negotiable student model is stored is given later. The interface for the enchanced version of EER-Tutor is shown in Figure 4. The most visible change is that the student model

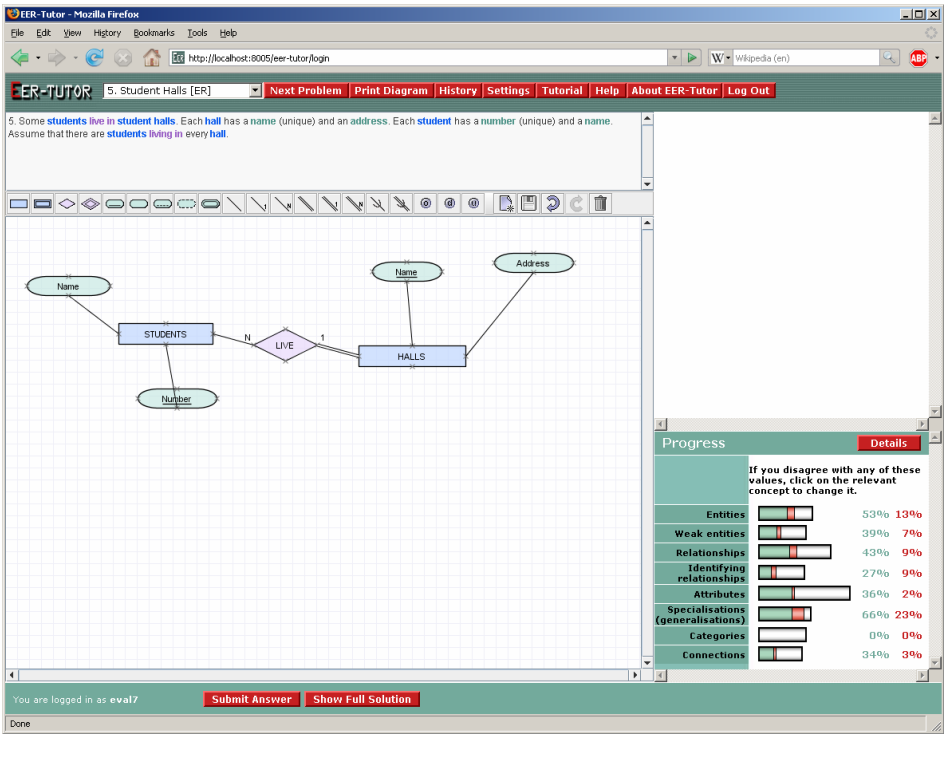

Figure 4. The enhanced EER-Tutor interface.

is now always displayed to the student. This was deliberate, as it should prompt the student to think more about their learning, thus increasing their meta-cognitive processes.

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. It is assumed that the student has used the system for some time before attempting to negotiate their model, and therefore will understand the concept names. Once the student initiates a dialog on a concept, the system asks the student a question relating to that concept, which, if they answer correctly, will increase their correct understanding knowledge component. The system currently supports two types of questions: multi-choice and short answer. If the student fails to answer the question correctly in a specified number of attempts, the system will decrease the correct understanding for that particular concept.

In the session illustrated in Figure 4, the student might believe that he/she knows more about Identifying relationships than currently shown in his/her student model. After clicking on that concept, the student is asked a question (illustrated in Figure 5). The system first asks the student to complete a statement about weak entities. In this particular situation, the student's answer is incomplete; the student answered "key", while the correct answer is "partial key". The system points out the mistake, and gives the student another chance to provide the answer. The student makes another mistake ("unique key"), which the system takes as evidence the student model should be negatively modified.

When using the negotiable student model, it is only possible to change the correct and incorrect components of skill meters, but not material covered. At best, the student 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 focus of the student; they still need to work on domain problems to progress through the tutor. A student who cannot solve problems will not be able to progress through EER-Tutor, even if they can correctly answer questions while negotiating their model.

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 6. Figure 7 shows the actual representation for question number 3, which features in the dialogue in Figure 5.

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

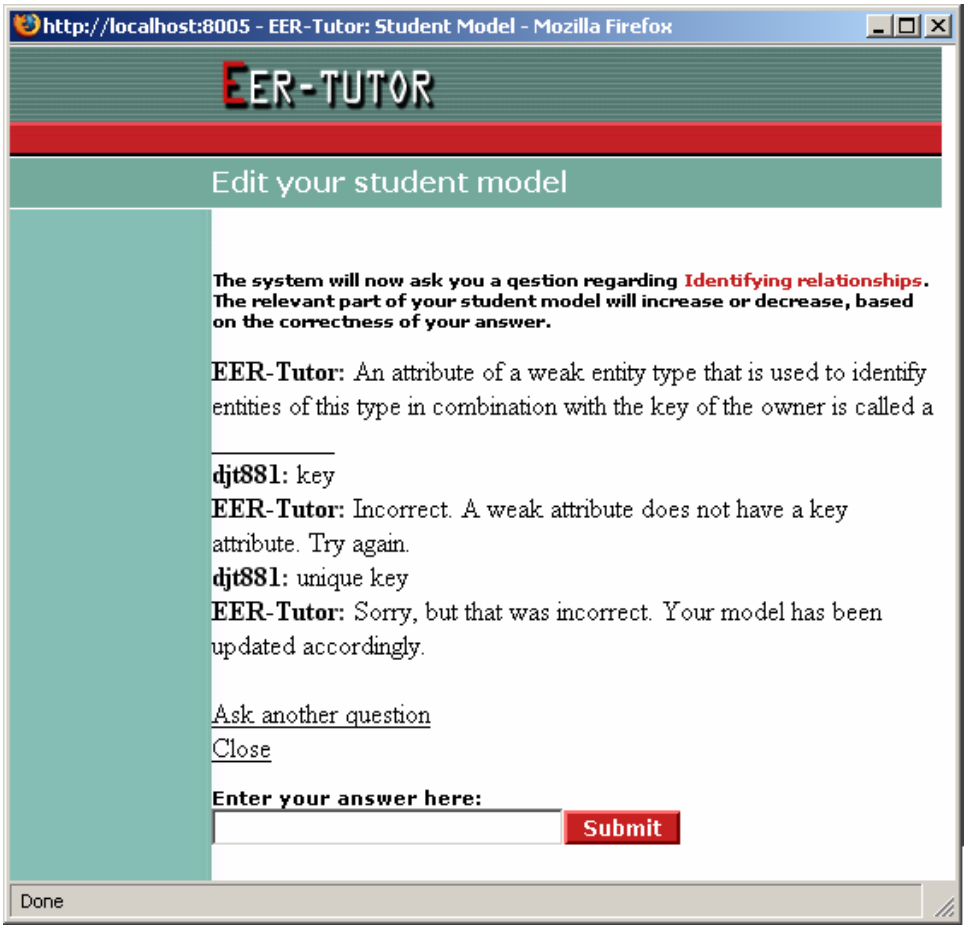

Figure 5. A short answer question.

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

Figure 6. General question structure.

one of these incorrect answers, the corresponding feedback will be displayed (as in Figure 5). As of writing, 48 questions have been defined, with a mix of multi-choice and short answer. Before further evaluation is conducted more questions need to be added.

```
(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 7. Question 3.

When the student initiates a dialogue a simple algorithm is used to decide which question to ask the student. The first preference is a question the student has not attempted before. If the student has attempted all the questions on the selected concept, the system will look for a question the student has attempted, but not answered correctly. If the student has answered all the questions on the concept correctly, a random question is selected. The selected question is always relevant to the selected concept, even if it means asking the student a question they have already answered correctly.

Figure 8 shows an example of a negotiable student model, after a period of use. For each concept a numerical value is stored, representing the outcome of the questions the student has answered on that concept. If this number is negative it means the student has answered more questions incorrectly than correctly. Each time the student answers a question, the value for the corresponding concept is incremented by positive or negative 0.01. This equates to a visible change of 1% when viewing the model. When combining the models to display to the student, this value is added to the correct knowledge component of the regular student model.

```
((Identifying relationships" 0.0)
 (Relationships" -0.07)
 (Entities" 0.15)
 (Attributes" -0.02)
 (Weak entities" -0.01)
 (Specialisations (generalisations)" -0.02))
(3 44 28 27 26 25 24 2 1 32 35 34 33 31 30 29 41
43 42 15 7 17 14 13 10 5 6)
(32 31 28 27 26 35 34 33 15 14 14 13 10 7 6 6 4 4
25 24 3 3 2 30 29 17 14 13 10 1 6 6 6 6 6 6 5 17
32 2 2 25 26 27 27 2 28 28 27 2 27 28 27 25 27 6
6 4 4 8 3)
```

Figure 8. A negotiable student model.

A list of all questions answered correctly, and a list of all questions attempted is also stored. This is used when choosing which question to ask the student.

5. Evaluation

We have performed a preliminary study of the negotiable student model in EER-Tutor, involving a subjective survey seeking opinions from students concerning the negotiable student model. The purpose of the evaluation was to gauge students' acceptance of a negotiable student model. Would they use it? Did they think it was a good idea? Did they like it? An important aspect of learning is motivation, so the subjective opinions of learners are crucial to any possible learning benefit.

Eleven postgraduate students from the Computer Science department of the University of Canterbury volunteered to participate in the study. Three participants had previously been involved in the process of developing earlier versions of EER-Tutor, and are referred to as experts. The participants were asked to use the system for an undetermined period of time, until they had a good feel for the negotiable student model. 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.

On average, the participants used the system for 42 minutes, with actual session lengths ranging from 5 to 157 minutes. The participants answered the total of 115 questions, at an average of 10.5 questions each. The three experts spent 35, 12, and 157 minutes using the system, and answered 10, 4, and 6 questions respectively. The 11 participants answered a total of 45 domain problems, giving an average of 4.1 problems each.

Table 1 shows the average rankings of aspects of EER-Tutor with the negotiable student model (NSM). The standard deviations are 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. Almost all the students felt the negotiable student model would be beneficial to learning, and many found it was an interesting challenge. Some students noted the negotiable student model provided something different to do for a while (i.e. another type of activity in addition to problem solving), and increased their motivation as it made using the system less

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)

repetitive. This in itself is a useful result as motivated students are likely to gain more from the system. The expert ratings were not consistently different to the non-expert ratings, with some values being higher and some lower than the overall average.

The next step is to perform an objective evaluation. For this, students will be split into two groups; one using the regular version of EER-Tutor, and one using the enhanced version with the negotiable student model component. Pre- and post-tests will be used to measure the learning gains achieved by students. Any differences between the two groups will be used to evaluate the effectiveness of the negotiable student model.

6. Conclusions

This project designed and implemented a negotiable student model in EER-Tutor. Much research has been done on open student models (Bull, 1997; Bull & Kay, 2007; Lazarinis & Retalis, 2007), but no negotiable student model has yet been implemented and evaluated in a large-scale Intelligent Tutoring System. Previous research (Bull & Pain, 1995; Kerly & Bull, 2008) has implemented a negotiable student model in only simple computer-based learning systems providing multi-choice quizzes; this project uses EER-Tutor, a complete ITS. EER-Tutor presents a problem-solving environment that the student must use to solve problems, and has a complex solution evaluator that gives dynamic feedback based on the student's actions within the system (Mitrovic *et al.*, 2007; Suraweera & Mitrovic, 2004). 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.

While using this enhanced version of EER-Tutor, students can negotiate their model, in addition to regular problem solving. While problem solving, the student can monitor their progress via their student model. If they feel their student model is an incorrect representation of their knowledge, the student can initiate a dialog with the system. The system asks the student a question on the selected concept, and modifies the student's model based on the correctness of the student's answer. This allows the student to modify their model, but they must first prove their knowledge to the system.

We evaluated the negotiable student model for EER-Tutor subjectively. All participants gave positive feedback and found the new component easy to use. We plan to conduct a controlled study so that the negotiable model can 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.

We designed the negotiable student model 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, which is another avenue for future research.

An interesting improvement would be to link the negotiable student model to constraints. Incorrect knowledge in a student model means that the student has violated a constraint. It would be possible to ask the student a question based on the constraint they violated, and answering it correctly would effectively satisfy the constraint, therefore removing the incorrect knowledge from the student model. Currently, students are asked a question at random, although it must be regarding the same concept. This sounds promising, but we must remember what the negotiable student model is for, that is, to encourage meta-cognitive processes. The purpose is not just to correct the holes in the students' knowledge, but rather to help them think about their learning, perform reflection, and other meta-cognitive processes. This improvement could still be implemented and evaluated, but we must be careful to keep the true purpose of this component in mind.

A questioned raised by this research is what is the best way to encourage users to engage in meta-cognitive processes and improve their meta-cognitive skills. The purpose of a negotiable student model is to develop these meta-cognitive skills, but how to best do this is still unknown. It is very likely that the simple question-answer approach used in this research is not the most effective way. Now that we have the framework in place, we can begin to experiment with other approaches to teaching meta-cognitive processes. Such research would be of significance to the educational community.

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

Aleven, V., & Koedinger, K. (2000). Limitations of student control: do students know when they need help? In G. Gauthier, C. Frasson, K. VanLehn (Eds.). *Intelligent tutoring systems: 5th int. conf.* (pp. 292–303). Berlin: Springer.

Bauer, M., Gmytrasiewicz, P. J., & Vassileva, J. (2001). Supporting Negotiated Assessment Using Open Student Models. Proc. UM 2001, LNAI 2109, pp. 295–297.

Beck, J., Stern, M., & Haugsjaa, E. (1996). Applications of AI in Education. *Crossroads*, *3*(1), 11–15,

Bennet, F. (1999). *Computers as tutors: solving the crisis in education*. Faben.

Bielaczyc, K., Pirolli, P., & Brown, A. L. (1993). Training in self-explanation and self-regulation strategies: Investigating the effects of knowledge acquisition activities on problem-solving. *Cognition and Instruction*, *13*(2), 221–252.

Bloom, B. S. (1984). The 2 sigma problem: The search for methods of group instruction as effective as one-to-one tutoring. *Educational Researcher*, *13*, 3–16.

Brusilovsky, P., Schwarz, E., & Weber, G. (1996). ELM-ART: An intelligent tutoring system on World Wide Web. In C. Frasson, G. Gauthier, A. Lesgold (Eds.). *3rd int. conf. intelligent tutoring systems* (pp. 261–269). Berlin: Springer Verlag.

Bull, S. (1997). See yourself write: A simple model to make students think. In A. Jameson, C. Paris and C. Tasso, (Eds.). *User-modeling: Proc. 6th int. conf. (UM97)* (pp. 315–326). New-York: Springer.

Bull, S., Cooke, N., & Mabbott, A. (2007). Visual attention in open learner model presentations: An eye-tracking investigation. In C. Conati, K. McCoy & G. Paliouras (Eds.). *User modeling 2007: 11th int. conf.* (pp. 187–196). Springer-Verlag, Berlin Heidelberg.

Bull, S., Dimitrova, V., & McCalla, G. (2007). Preface for special issue (part 1) open learner models: Research questions. *Int. J. Artificial Intelligence in Education*, *17*, 83–87.

Bull, S., & Kay, J. (2007). Student models that invite the learner, in: The SMILI open learner modelling framework. *Int. J. Artificial Intelligence in Education*, *17*, 89–120.

Bull, S., & Pain, H. (1995). 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).

Dimitrova, V. (2003). STyLE-OLM: Interactive open learner modelling. *Int. J. Artificial Intelligence in Education*, *13*, 35–78.

Kerly, A., & Bull, S. (2008). Children's Interactions with Inspectable and Negotiated Learner Models. In B.P. Woolf, E. Aimeur, R. Nkambou & S. Lajoie (Eds.). *Intelligent tutoring systems: 9th int. conf.* (pp. 132–144). Springer-Verlag, Berlin Heidelberg.

Lazarinis, F., & Retalis, S. (2007). Analyze me: Open learner model in an adaptive web testing system. *Int. J. Artificial Intelligence in Education*, *17*, 255–271.

Mathews, M., Mitrovic, A., & Thomson, D. (2008). Analyzing high-level help seeking behaviour in ITSs. In W. Nejdl *et al.* (Eds.), *AH 2008*, LNCS 5149, pp. 312–315.

Mitrovic, A., & Martin, B. (2007). Evaluating the effect of open student models on self-assessment. *Int. J. Artificial Intelligence in Education*, *17*, 121–144.

Mitrovic, A., Martin, B., & Suraweera, P. (2007). Intelligent tutors for all: Constraint-based modeling methodology, systems and authoring. *IEEE Intelligent Systems*, *22*(4), 38–45.

Parker, L. (2008). Little wonders. *Australian Educator* (pp. 18–20). (Spring 2008).

Self, J. A. (1990). 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.

Suraweera, P., & Mitrovic, A. (2004). An intelligent tutoring system for entity relationship modeling. *Int. J. Artificial Intelligence in Education*, *14*(3–4), 375–417.

Swanson, H. L. (1990). Influence of metacognitive knowledge and aptitude on problem solving. *J. Educational Psychology*, *82*, 306–314.

Van Labeke, N., Brna, P., & Morales, R. (2007). Opening up the interpretation process in an open learner model. *Int. J. Artificial Intelligence in Education*, *17*, 305–338.