Evaluating the Effects of Open Student Models on Learning

Antonija Mitrovic
Brent Martin

Intelligent Computer Tutoring Group
Computer Science Department, University of Canterbury
Private Bag 4800, Christchurch, New Zealand
tanja@cosc.canterbury.ac.nz
bim20@cosc.canterbury.ac.nz

Abstract: In previous work [10], we reported on an experiment performed in the context of SQL-Tutor, in which we analysed students' self-assessment skills. This preliminary study revealed that more able students were better in assessing their knowledge. Here we report on a new study performed on the same system. This time, we analysed the effect of an open student model on students' learning and self-assessment skills. Although we have not seen any significant difference in the post-test scores of the control and the experimental group, the less able students from the experimental group have scored significantly higher than the less able students from the control group. The more able students who had access to their models abandoned significantly less problems the control group. These are encouraging results for a very simple open model used in the study, and we believe that a more elaborate model would be more effective.

1 Introduction

Self-assessment is one of the meta-cognitive skills necessary for effective learning. Students need to be able to critically assess their knowledge in order to decide what they need to study. The same skill is also important for students to assess the difficulty of the problem they are working on, and to decide whether to abandon to problem or keep working on it. Intelligent educational systems must support the acquisition of meta-cognitive skills in order to support deep learning, and therefore must also support the acquisition of self-assessment skills.

In previous work [10], we hypothesized that more able students were better when assessing their own knowledge than their less able peers. This hypothesis is in accordance with findings from other studies [1]. In order to evaluate the hypothesis, we analysed students' self-assessment skills in the context of the SQL-Tutor system. The system was modified slightly to allow for data collection. We focused on situations when students abandoned the problem they were working on, and asked for a new problem. In such situations, students were asked two questions. Firstly, we asked the student to specify the reason for abandoning the current problem. Three possible replies were offered: the student may think that the current problem was too easy or too difficult, or may simply want to work on a problem of a different nature. The student was then asked to specify what kind of problem they would like to work on next. For this purpose, problems were characterized by the clause, so seven options were available, one for each clause of the SELECT statement, plus the *any clause* option. The results of the data analyses performed supported our hypothesis.

In this paper, we report on a new experiment conducted in the context of the same system. We were interested in using open student models as a way to support self-assessment skills. We begin by overviewing related work in Section 2, and then introduce SQL-Tutor briefly in Section 3. SQL-Tutor provides a facility for students to select problems on their own, which requires students to be able to assess their own knowledge. Section 4 describes the experiment performed, and is followed by a description of the findings in section 5. The conclusions are presented in the final section.

2 Related Work

Metacognition includes the processes and activities involved with awareness of, reasoning and reflecting about, and controlling one's cognitive skills and processes. A number of studies showed that better metacognitive skills result in improved problem solving and better learning [2, 6, 7, 15, 16], and that such skills can be taught [3, 6].

White et al. [16] define metacognition as consisting of three fundamental components*: knowledge about knowledge* (including cognitive and social expertise), *regulatory skills*, needed to monitor and control knowledge and *development expertise*, which allows students to reflect and improve knowledge. Their hypothesis is that students need to be conscious of theories for socio-cognitive processes for learning. In [16] they present SCI-WISE, a system that supports inquiry learning. In this type of learning, students cycle through several stages, starting by formulating a question, followed by the generation of hypotheses, and experimentation. After that, collected data is analysed, and students develop a model on the basis of results of analysis, and finally evaluate the model by applying it in a novel situation. The system supports inquiry learning by asking the student to explicitly state the purpose of each action and provide a justification for it. The system provides a simulation environment in which to perform experiments, and incorporates several agents, each of which specializes in a particular metacognitive process, such as inventing, collaborating and analysing, and advises the student in relevant situations. The student develops his/her own theory of scientific inquiry by modifying the agents so that they reflect student's beliefs. In order to test his/her theory, the student then engages in a physics project, during which agents provide advice. The student evaluates the agents' behaviour, which may result in the modification of the theory.

Most of the metacognitive research focuses on self-explanation, which is a skill of "generating explanations and justifications to oneself to clarify an example solution" [2, 7]. Self-explanation enables students to focus on general principles by examining specific examples [6]. Aleven and Koedinger [1] evaluate students' abilities to identify situations when help is needed and to ask for appropriate help. They show that not all students possess this skill, and recommend several ways in which the system may support students in acquiring it.

Reflection is a metacognitive skill in which students examine their own knowledge. This metacognitive skill is encouraged by allowing the student to inspect and, in some cases, to modify the student model. Several projects focused on the effects of opening the student model on students' learning [4, 5, 8, 9]. The efforts range from simply visualizing the model, to actively involving the student in the modeling process through negotiation or collaborative construction of the model. In these systems, the student model is not just a source of knowledge about the student of value to the system, but becomes an important learning resource in its own right.

The representations used to visualize the student model range from simple skillometers to very complex ones. Skillometers allow for an easy to understand, high-level overview of the student model, and have been used in ELM-ART [4] and cognitive tutors [1]. In addition to just showing the student model at a high level, some systems allow the student to challenge and negotiate the content of the model. This process is referred to as open interactive [8], collaborative [5], cooperative [9] or participative [12] student modelling. Such approaches use more complex representations of the student model based on conceptual graphs [8], Bayesian networks [17], tree structures [9], tables [5] and Prolog clauses [14]. These projects differ in the content of the student model, representation chosen for visualizing the model, the type of interactions (inspection or negotiation) and the mechanism for dealing with conflicts. The area is still just emerging, and proper ways of evaluating developed systems is yet to be identified. In all projects only very limited evaluation has been done, the results of which are not irrefutable.

3 SQL-Tutor

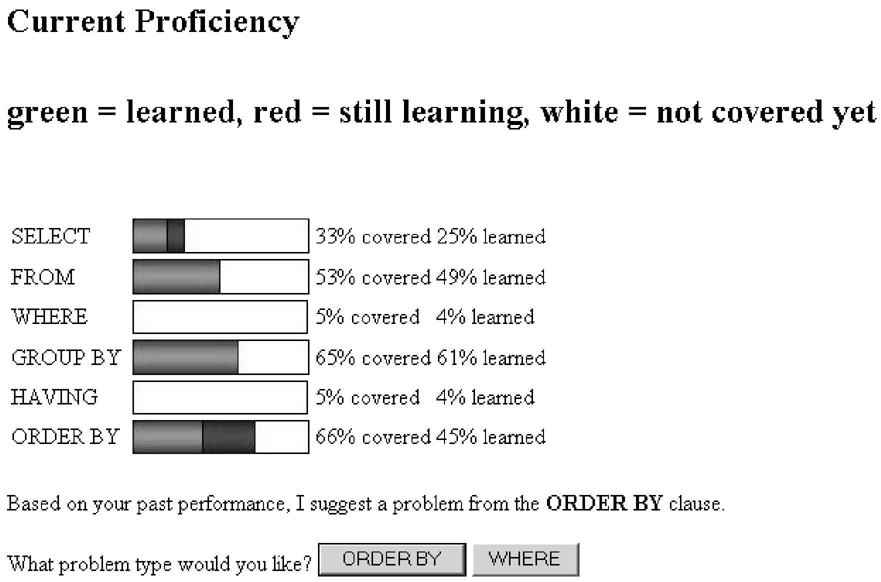
SQL-Tutor is an intelligent educational system aimed at university-level students learning SQL. For a detailed discussion of the system, see [11]; here we present only some of its features. SQL-Tutor consists of an interface, a pedagogical module, which determines the timing and content of pedagogical actions, and a student modeller, which analyses student answers. The system contains definitions of several databases, and a set of problems and the ideal solutions to them. SQL-Tutor contains no problem solver. To check the correctness of the student's solution, SQL-Tutor compares it to the correct solution, using domain knowledge represented in the form of more than 500 constraints. It uses Constraint-Based Modeling [13] to model knowledge of its students. Students may work their way through a series of problems for each database, or ask the system to select a problem on the basis of his/her student model.

4 The Study

The purpose of this study was to determine the effect of a simple open student model on students' learning and self-assessment skills. Let us first describe the way we visualize the student model. The student model in SQL-Tutor is implemented as an overlay on top of the constraint base. There are currently more than 500 constraints in the system, and therefore it is not possible to visualize information about each constraint. Instead, we have decided to compress the student model into a simple structure that resembles the structure of the SELECT statement. The student is shown six skillometers, which show the student model in terms of the six clauses of the SELECT statement. For each clause, we find all the relevant constraints, and compute the *coverage* (the percentage of constraints that the student has used) and *correctness* (i.e. the percentage of all relevant constraints that the student has used correctly). These two percentages are visualized as shown in Figure 1.

We focus on situations when students abandon the current problem and ask for a new one. In such cases, the students were asked the same two question as in the preliminary study: to specify the reason for abandoning the current problem, and to specify the type of the problem they would like to work on next. The available options were the six clauses of the SELECT statement (*SELECT, FROM, WHERE, GROUP BY, HAVING,* and *ORDER BY*), plus the *any clause* option. For the experimental group, if the student's selection of the problem type was different from the one suggested by the system, the student was shown the open model, and asked whether they wanted to continue with the type of problem they specified, or whether they would prefer to switch to the system's suggestion.

The participants were enrolled in an introductory database course at the University of Canterbury, New Zealand, in the second half of 2001. Prior to the experiment, all students listened to four lectures on SQL and had two labs on the Oracle RDBMS. During the experiment, there were 4 additional lectures on SQL, and a series of four



Current Proficiency

green = learned, red = still learning, white = not covered yet

SELECT 33% covered 25% learned
FROM 53% covered 49% learned
WHERE 5% covered 4% learned
GROUP BY 65% covered 61% learned
HAVING 5% covered 4% learned
ORDER BY 66% covered 45% learned

Based on your past performance, I suggest a problem from the **ORDER BY** clause.

What problem type would you like?

Fig. 1. The visualization of the student model

more labs. The experiment required the students to sit a pre-test, which was administered in a lecture on 10 September 2001. The pre-test consisted of three multichoice questions. All three questions contained the text of the query, and some solutions. The students were asked to classify the solutions as correct or incorrect. The maximum mark for the pre-tests was 9.

The students who sat the pre-test were given user accounts to use in SQL-Tutor from September 12, and were randomly allocated to one of the three possible versions of the system. The experiment was designed to combine two evaluations into one: the evaluation of an enhanced constraint and problem set, and the evaluation of the open student model. The control group served as the control group for both studies. The problem selection group is not of importance for this paper, and the experimental group is the group that had access to the open student model.

The interaction with the system was voluntary. The course involved a test on SQL a month after the system was introduced to the class. The post-test consisted of three questions of similar nature and complexity as the questions in the pre-test, and was administered in a lecture, on 9 October 2001.

5 Results

This section presents the results of the analyses performed on the data collected in the experiment. Section 5.1 presents the general findings about how students learnt with SQL-Tutor. Data analyses relevant to our hypotheses are discussed in Section 5.2.

5.1 Learning with SQL-Tutor

Out of 159 students enrolled in the course, 100 sat the pre-test. Table 1 gives the number of students in each group, their pre-test scores, and some additional information about their logs. The mean score for the pre-test for the whole class was 5 (out of 9, SD=1.36). The t-test reveals there are no significant differences between the pre-test scores, which means that the three groups are comparable. The problem selection group is of no importance for this paper, so we report on the control and experimental groups only.

As the usage of the system was voluntary, 80 students actually logged on to SQL-Tutor. Table 1 gives the number of students in each of the groups who sat the pre-test, and also the number of students who actually used the system. However, some of these students have only briefly looked at the system. We excluded the logs of 9 students who attempted no problems, and the number of valid logs is given in the table.

Group	Students	Pre-test mean (SD)	Accounts used	Valid logs	Post-test
Control	34	4.82 (1.44)	29	24	12
Experimental	33	5.12 (1.41)	23	21	12
Problem selection	33	5.06 (1.25)	28	26	14

Table 1. The three groups

There were 101 students who sat the post-test (mean=5.99, SD=1.60). In the post-test, the students were asked whether they have used SQL-Tutor and, if they have, to specify their account. The mean score on the post-test for students who have used SQL-Tutor was 6.42 (SD=1.50), which is higher than the overall mean. The mean score of the students who have not used SQL-Tutor was 5.67 (SD=1.61). However, we cannot claim that SQL-Tutor was responsible for this higher mean, as the participation in the study was voluntary, and the students who participate are usually more motivated.

Although we asked students to specify the account if they used the system, some students have not done that. The last column in Table 1 contains the number of students in each group who specified their account names. These are the only students whose pre- and post-test results we were able to match. Table 2 contains the statistics for the pre- and post-test results for those students only. It can be seen that the scores for the experimental group are slightly higher, but not significantly.

Group	Matched tests	Pre-test mean (SD)	Post-test mean (SD)
Control	12	4.42 (1.24)	6.42 (1.38)
Experimental	12	4.50 (1.17)	6.67 (1.56)

Table 2. Pre- and post-test results for the students who can be matched

Table 3[1] gives some simple statistics gathered from the valid logs. The number of sessions ranged from 1 to 13, and the lengths of individual sessions ranged from 1 minute to almost four hours. The total interaction time ranged from 5 minutes to more than ten hours. The *Total solved* column gives the mean number of problems the students have solved during all sessions, which ranges from 1 to 70. *Problem/session* gives the average number of problems students have seen in a session (ranges from 1 to 40). *Solved/session* gives the averages for the number of solved problems (the minimum for solved problems per session is 0, the maximum is 35), while the percentage of problem completion is given in the next column. On the average, the control group needed slightly more time per solved problem than the experimental group. None of the reported numbers are significantly different.

Group	Sessions	Session length (min)	Total time (min)	Total solved	Problem /session	Solved /session	% prob. solved /session	Mins per solved
Contr (24)	4.7 (3.28)	39 (41)	183 (189)	20.5 (15.2)	5.3 (3.1)	4.4 (3.3)	77.6 (21.7)	8.6 (5.2)
Exper. (21)	4 (2.66)	36 (32)	144 (125)	22.8 (18.8)	6.7 (3.8)	5.6 (3.7)	77.1 (18.2)	7.2 (3.7)

Table 3. Statistics about the sessions

[1] Standard deviations are given in parentheses.

5.2 Analyzing the Self-Assessment Skills

We also analysed the effects of the open student models with respect to students' abilities. Each group was split into two subgroups, with less and more able students, depending on their scores on the pre-test. Students who scored above the average (5 or more marks) on the pre-test were put into the *more able* group, while the students who scored 0 to 4 marks were put into the *less able* group. Table 4 contains statistics about the subgroups, which are of similar sizes. There is no significant difference on the pre-test scores between the less able and more able students in the control and the experimental groups. The results on the post-test for the more able students are not significantly different. However, the scores on the post-test for the less able students in the experimental group are significantly better than the results of the less able part of the control group (t=1.43, p<0.1). This result suggests that the less able students benefit more from open student models.

The logs also contain the data relevant to our hypothesis. Out of 45 logs we analysed, 12 students did not abandon any problems. These students interacted with the system for a shorter time (105 minutes on average) and solved 16 problems. However, they solved all the problems they attempted. The remaining 33 students abandoned some problems. Whenever a student asked for a new problem before completing the current one, the system asked the two questions. Each student was asked these questions at least once, and at most 40 times, and the means for the four subgroups are given in Table 4 in the *Questions* row. Out of the total of 242[2] abandoned problems, 93 (38.4%) were from the more able students, and 149 were from the less able students. Therefore, less able students are much more likely to abandon a problem. The more able students from the experimental group have abandoned significantly less problems than the more able students from the control group. However, there is no significant difference between the two groups of less able students. This might suggest that the more able students were encouraged to complete the problems by being exposed to their student models.

	Control		Experimental	
	Less able	More able	Less able	More able
Pre-test	3.6 (0.5)	5.6 (0.9)	3.7 (0.5)	5.6 (0.9)
Post-test	5.9 (1.5)	7.2 (0.8)	7 (1.5)	6.2 (1.6)
Time	164 (135)	321 (229)	169 (142)	115 (107)
Problems solved	19.8 (10.4)	28.8 (17.2)	27.4 (26.2)	19.4 (14.4)
Questions	7.1 (4.1)	10.87 (13.2)	8.5 (5)	3.9 (3)
0 attempts	5.1 (2.9)	7.6 (11.2)	4.7 (2.8)	2.1 (1.6)
0/new problem	1.6 (1.1)	2.2 (3.6)	1.1 (0.7)	1.5 (1.5)

Table 4. Statistics for the groups of students with different prior knowledge

[2] Although we use the word "problem" here, we refer to situations when the student abandons the current problem, and get the two questions. It may be the case that the student abandons the same problem several times, so the total number reported here does not equal to the number of distinct problems abandoned.

The number of attempts before abandoning the problem ranged from 0 (the total of 165 cases) to 13 (an average of 1.43). The *0 attempts* row in Table 4 gives the mean number of cases when students abandon the current problem without attempting it. The numbers reported here are a bit misleading, as they include all problems, not just the new problems. In other words, the student might attempt a problem, and then abandon it, only to receive the same problem again, and then abandon it for the second time. Such a problems is counted twice. The following row (*0/new problem*) counts each problem only once in such situations. It can be seen that the more able students in the experimental group attempt to solve problems more often than the more able students from the control group; however, the differences are not significant.

The distribution of answers to the first question is given in Figure 2. The control group students thought that the problem is too easy more often than experimental group students, especially the more able ones, although the inspection of the logs very often contradicts the reason they specified.

Figure 3 shows the distribution of answers to the second question. As we hypothesized, less able students are not good at identifying the kind of problem to work next, and therefore they specify *any clause* most often (in 54.69% of the cases in the control group, and 40.98% in the experimental group).

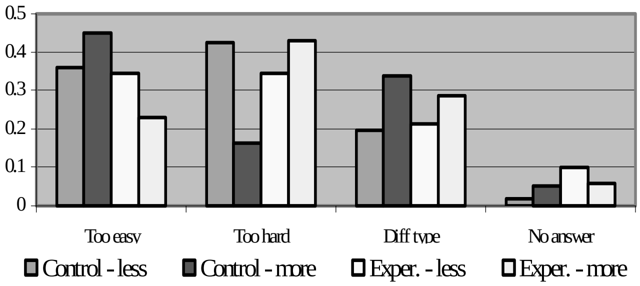


Fig. 2. The distribution of answers for question 1

After answering the second question, the students in the experimental group were shown the summary of their student models if their selection of a clause to work on next differed from the system's selection. The number of such cases for the two subgroups are given in the *No of cases* column in Table 5. In those situations, the students were asked to specify whether they wanted to go on with their selection, or adopt the system's suggestion. The following column gives the percentages of the total number of questions when the student's selection differed from the system's one. Although the more able students have opinions about what they should be doing that more often varies from the system's suggestion, compared to the less able students, the difference is insignificant. The percentages of cases when the student accepts the system's suggestion (last column in Table 5) is almost identical for the more and less able students. When comparing the experimental group with the control group, there is no significant difference in the percentages of cases when more able student's selections differ from system's selections.

Group	No of cases	% of questions	Student agrees	% agree
Exper. - more able	2 (2.69)	57.25 (42.57)	0.75 (0.70)	51.56 (51.94)
Exper. - less able	3.70(3.40)	35.78 (31.70)	1.80 (1.64)	57.5 (42.94)
Control – more able	8.25 (11.37)	42.20 (28.20)	N/A	N/A
Control – less able	5.44 (4.44)	41.20(20.62)	N/A	N/A

Table 5. The statistics for the two subgroups of the experimental group

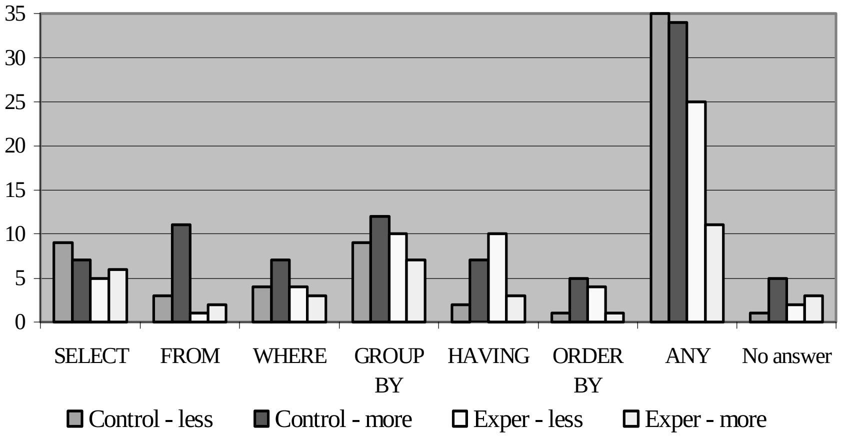


Fig. 3. The distribution of answers for Question 2

6 Conclusions

We presented a study of the effects of open student models on student self-assessment skills. Although we have not seen any significant difference in the post-test scores of the control and the experimental group, the less able students from the experimental group have scored significantly higher than the less able students from the control group. The more able students who had access to their models abandoned significantly less problems than the more able students from the control group, and had stronger opinions on what they should work on next, which often varied from the system's suggestions.

Overall, these results suggest that the open model may have improved the performance of the less able students, and that it may have boosted the self-confidence of the more able students, such that they abandoned fewer problems and judged their own abilities more readily. Alternatively, more able students in the experimental group may have specified a clause more often *in order to view the model*, which would suggest that they judged it to be beneficial. Such results are encouraging given the simplicity of the open student model developed for this study. Our students only received a high-level overview of their knowledge, and they could not challenge it or make any modifications. We believe that a more sophisticated open student model would be more effective, and plan to conduct a new study along these lines in the future.

Acknowledgements: The work presented here was supported by the University of Canterbury grant U6430.

References

1. Aleven, V., Koedinger, K. (2000) Limitations of Student Control: Do Students Know When They Need Help? Proc. ITS'2000, Springer-Verlag, pp. 292-303.
2. Aleven, V., Koedinger, K., Cross, K. (1999) Tutoring Answer Explanation Fosters Learning with Understanding. In: Lajoie, S.P., Vivet, M. (eds): Proc. AIED'99, pp. 199-206.
3. Bielaczyc, K., Pirolli, P., Brown, A.L.: Training in Self-Explanation and Self-Regulation Strategies: Investigating the Effects of Knowledge Acquisition Activities on Problem-solving. *Cognition and Instruction*, 13(2) (1993) 221-252.
4. Brusilovsky, P., Schwarz, E., Weber, G. (1996) ELM-ART: an Intelligent Tutoring System on World Wide Web. Proc. ITS'96, pp. 261-269.
5. Bull, S., Brna, P. (1997) What does Susan Know that Paul Doesn't? (and vice versa): Contributing to each other's student model. Proc. AIED 97, pp. 568-570.
6. Chi, M. T. H., Bassok, M., Lewis, M.W., Reinmann, P., Glaser, R. (1989) Self-Explanations: How Students Study and Use Examples in Learning to Solve Problems. *Cognitive Science*, 13, 145-182.
7. Conati, C., VanLehn, K.: Further Results from the Evaluation of an Intelligent Computer Tutor to Coach Self-Explanation. Proc. ITS'2000, Springer-Verlag, (2000) 304-313.
8. Dimitrova, V., Self, J., Brna, P. (2001) Applying Interactive Open Learner Models to Learning Technical Terminology. Proc. UM 2001, Springer, pp. 148-157.
9. Kay, J. (1995) The UM toolkit for cooperative user modelling. *User Modelling and User-Adapted Interaction*, 4, 149-196.
10. Mitrovic, A. Investigating students' self-assessment skills. Proc. UM-2001, Sonthofen, July 2001, Springer-Verlag LNAI 2109, pp. 247-250.
11. Mitrovic, A., Ohlsson, S.: Evaluation of a Constraint-based Tutor for a Database Language. *Int. J. on Artificial Intelligence in Education*, 10(3-4), (1999) 238-256.
12. Morales, R., Pain, H., Conlon, T. (2001) Effects of Inspecting Learner Models on Learners' Abilities. Proc. AIED 2001, pp. 434-445.
13. Ohlsson, S.: Constraint-based student modeling. In: Greer, J.E., McCalla, G (eds): *Student modeling: the key to individualized knowledge-based instruction*, (1994) 167-189.
14. Paiva, A., Self, J. (1995) TAGUS – a user and learner modelling workbench. *User Modeling and User-Adapted Interaction*, 4, 197-226.
15. Swanson, H.L.: Influence of Metacognitive Knowledge and Aptitude on Problem Solving. *J. Educational Psychology*, 82 (1990) 306-314.
16. White, B.Y., Shimoda, T. A, Frederiksen, J.R. (1999) Enabling Students to Construct Theories of Collaborative Inquiry and Reflective Learning: Computer Support for Metacognitive Development. Int. Journal on AI in Education, vol 10, 151-182.
17. Zapata-Rivera, J.D., Greer, J.E. (2000) Inspecting and Visualizing Distributed Bayesian Student Models. Proc. ITS'2000, Springer, pp. 544-553.