

Supporting Learning by Opening the Student Model

Danita Hartley and Antonija Mitrovic

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

Abstract: Intelligent tutoring systems (ITSs) provide individualised instruction by maintaining models of their students. Traditionally, these models have been hidden from the student. However, recent work in the area has suggested educational benefits in exposing the student model. This approach, known as open student modelling, allows the student to inspect their model thereby facilitating reflection, which is known to enhance the learning process. To date, few evaluations have been conducted to determine the effects that open student models have on learning. This is the focus of our work. In particular, we are interested in whether even a simple open model can have a positive effect on learning. We have exposed the student model in e-KERMIT, and performed an evaluation study. Subjective results from the study are encouraging, although a more extensive study is needed to draw reliable conclusions.

1. Introduction

Developing models of students' knowledge has been one of the central research topics in the area of ITS. Until recently, the learners have not been aware of the existence of student models. This situation has begun to change in the previous decade. Several projects focused on the effects of opening the student model on students' learning. 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. The intention is to engage the student in thinking about his/her own knowledge, thus involving the student at the meta-cognitive level. The student model is not just a source of knowledge about the student of value to the system, but becomes an important learning resource on its own. Students who engage on the meta-cognitive level achieve significantly better results than students who do not [19].

In this paper we present our experiences in opening the student model of KERMIT [18], an ITS for database design. A high-level overview of the student model (SM) is always visible, but the student can explore the model in detail. The paper is structured as follows. Section 2 presents related work. KERMIT is briefly introduced in section 3, and its extension (e-KERMIT) is presented next. We

performed an experiment described in Section 5, and the results of data analyses are given in the following sections. The conclusions are given in the final section.

2. Related Work

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 are used in ELM-ART [2] and cognitive tutors [1]. Some systems additionally allow the student to challenge and negotiate the content of the model. This process is referred to as open interactive [9], collaborative [6], cooperative [11] or participative [15] student modelling. Such approaches use more complex representations based on conceptual graphs [9], Bayesian networks [20], tree structures [11], tables [3, 4] and Prolog clauses [17].

TAGUS [17] is a learner modelling system that allows external agents (the student or an educational system) to inspect and modify the SM. The model is represented as a set of Prolog clauses, which makes communication with the student quite complex. The student can insert new information into the SM, and also can update, delete or revise it. Belief revision techniques are used to maintain the model.

The **um** toolkit [11] is a shell for building and inspecting student models. The student and the system maintain their private models, but can also share some information. The SM contains learner's preferences, attributes and beliefs. The student may inspect the model, ask for justifications of system's beliefs and modify them. The toolkit has been used to demonstrate a system that teaches students to use a text editor. 30% of students involved in a study inspected their student models, but more than half of the students did not show significant interest in the model.

Collaborative modelling allows the student to develop his/her model jointly with the system and maintain dialogues about its content. Mr. Collins [4, 5] teaches the use of pronouns in Portuguese, using a small knowledge base of 12 rules only. The student model consists of system's beliefs induced from the observed behaviour, and the student's beliefs, which are explicitly specified by the student. For each rule, the system maintains statistics of how the student has used it, and the student also specifies his/her confidence that he/she knows the rule. If the student's belief is the same or within one level of the system's belief, the student model is treated as correct. However, if the two beliefs vary significantly, the system asks the student to revisit his/her confidence, and presents recent attempts at the rule with justification of its own belief. The student may disagree with the system, and may need to answer additional questions in order to justify his/her confidence. The evaluation study shows that students are willing to inspect their SM and challenge it. However, the study involved only 9 participants, and the results are therefore not irrefutable.

[15] describes a learning system in which the student is to control a pole on a cart in a simulated environment. The student can inspect and modify the SM, and has to articulate the strategies used when working in the environment. The study revealed positive effects of the open SM on reflection. However, although the students showed improved awareness, their performance on a transfer task was poor.

StyLE-OLM [9] is an open learner modelling component of a system for teaching technical terminology in a foreign language. The student model is represented as a conceptual graph. Both the student and the system can question the SM, introduce or withdraw prepositions and justify claims. The system uses the belief modal logic and complex inference mechanism to produce a representation on which both sides agree. A limited study (with seven participants) revealed that more able students are better in reflecting on their knowledge. ViSMod [20] is a similar system that allows the student to create his/her model in terms of a Bayesian network.

Finally, several research projects focus on peer diagnosis as a way to involve students in reflection. PeerSM [6], PairSM [7], Phelps [10] and ConceptLab [21] provide collaborative environments where groups of students discuss the content of SM. PeerISM [8] is an extension of this idea, where an artificial peer monitors the modelling process and advises on conflicting points. I-Help [12] is another system based on peer help, but learner models are distributed over various resources. Every learner in I-Help has his/her own agent, the goal of which is to contact suitable peers and negotiate with their agents about help that is needed. Thus, every agent needs to model his/her own students, but also other students in the system.

The above projects differ in the content of the student model, representation chosen for visualization, 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.

3. KERMIT – a Database Design Tutor

KERMIT (Knowledge-based Entity Relationship Modelling Intelligent Tutor) is an ITS aimed at the university-level students learning conceptual database design. The architecture of the system is illustrated in Figure 1. For a detailed discussion of the system, see [18]; here we present some of its basic features. KERMIT is a problem-solving environment in which students can practice database design using the Entity Relationship (ER) data model. The system consists of an interface, a pedagogical module, which determines the timing and content of pedagogical actions, and a constraint-based modeller, which analyses student answers and generates student models.

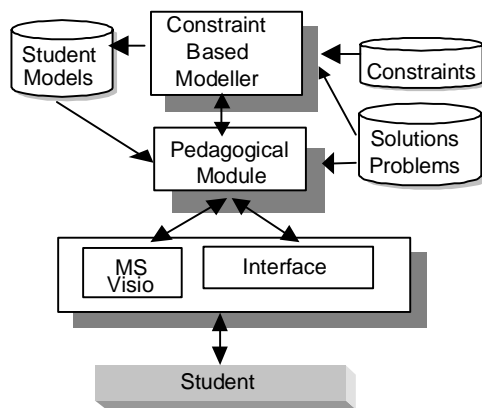


Fig. 1. The architecture of KERMIT

KERMIT contains a set of problems and ideal solutions, but has no problem solver. In order to check the correctness of the student's solution, KERMIT compares it to the correct solution, using domain knowledge represented in the form of more

than 90 constraints. It uses Constraint-Based Modeling [16] to model knowledge of its students. The constraints cover both syntactic and semantic knowledge. An example of a syntactic constraint is "An entity cannot be directly connected to another entity". Semantic constraints relate the student's solution to the system's ideal solution, testing whether the student has modelled all requirements. There are constraints that check for equivalent, but not identical ways of modelling parts of a database in the student's and ideal solution. One of such constraints deals with multi-valued attributes of entities, which may alternatively be modelled as weak entities.

The interface is composed of three windows tiled vertically (see Figure 2). The top window displays the current problem and provides controls for stepping between problems, submitting a solution and selecting feedback level. The middle window is the main working area. In this window the student draws ER diagrams using the toolbar on the left side of the window. Feedback is presented in the lowest window in the textual form, as well as through an animated pedagogical agent.

4. e-KERMIT

As stated previously, our goal is to determine whether an open SM has an effect on student's learning. The systems presented in [9] and [21] are not full educational systems; in both cases only one isolated component has been developed. Conceptual graphs and Bayesian networks are used to represent domain knowledge, and also to represent the SM. Although such a decision is logical in these two cases, this is not possible in all situations. In the case of KERMIT, the knowledge base consists of a large number of constraints that are not related to each other. The constraints may be quite complex, and it is therefore not possible to visualize the SM in the same form as the underlying domain knowledge. Another difficulty comes from the size of the knowledge base, which is much bigger than in the mentioned systems. Therefore, we have decided to use a different representation. The open SM in e-KERMIT is a summary of the real SM, illustrated in a form of a hierarchy. The constraints are grouped according to the pedagogically important domain categories, while the individual constraints appear only as leaves in the hierarchy.

Following that, we modified KERMIT's interface to include the open SM, as shown in Figure 2. The only difference from the original interface is that the feedback window (the lowest part of the interface) is now divided into two parts. The left part of the window includes a summary of the student's progress, the number of problems the student has completed, the option of requesting the detailed view of their progress (the *Show Me More* button), and access to a tutorial on understanding the main progress view (the *Help* button). Summary statistics are shown because the main progress view is too large to display with the main interface. The summary of the SM provides constant feedback on progress and acts as an aid to remind and motivate students to further inspect their models. Figure 3 depicts the window that shows the hierarchical view of a student's progress, with some of the categories expanded. The window is split into two frames. The top frame displays the hierarchical taxonomy of ER knowledge, with progress statistics for each category. For example, consider the *Attribute Identification* category in Figure 3. The fraction to the right of the progress

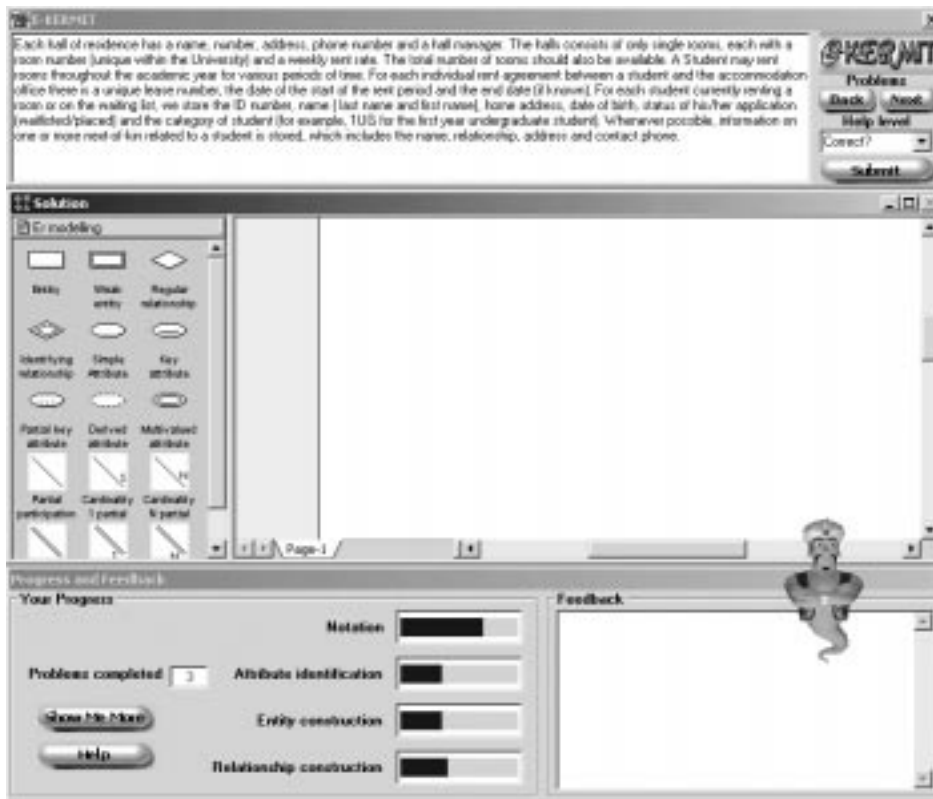


Fig. 2 The interface of e-KERMIT

bar shows the student's score (33%) out of the percentage of material for attribute identification covered so far (44%). Besides summarizing the student's knowledge, the open model also presents a high level view of the domain, which supports student's understanding of the domain structure. By inspecting their models, students reflect on their knowledge, reconsider their beliefs about the domain concepts.

The lower frame provides a textbox that describes the currently selected category. This is an aid to provide an explanation for any part of the taxonomy that the student may have problems interpreting. In addition, the first time a student accesses their progress they are given a short tutorial on how to interpret what they see. The tutorial is made available at any time by clicking the *Help* button in the summary progress view in the main interface. The aim of the tutorial is to facilitate a quick understanding by giving students an introduction to how their progress is represented, and what the percentages convey.

5. The Experiment

An evaluation study was carried out with students enrolled in an introductory database course at the University of Canterbury. The experiment was run during the normal lab hours over the duration of one week. The evaluation focused on whether students learn more with an open SM, whether they inspect the models and feel that

the open SM contributed to their learning. We were also interested in any differences between more and less able students with respect to the above.

The students were assigned to one of two groups: A control group interacted with the original KERMIT system, and the experimental group interacted with e-KERMIT. The groups were assigned to different rooms to prevent students from being exposed to both systems. Each student participated in a single session of up to 110 minutes. Data collection consisted of four stages: pre-testing, system interaction, post-testing and subjective system assessment. Since participation was voluntary and laboratory attendance varies, it was hard to control group sizes. The pre/post tests consisted of three questions each, of similar difficulty. To minimise any effect resulting from variation in test difficulty, the tests were rotated between successive sessions.

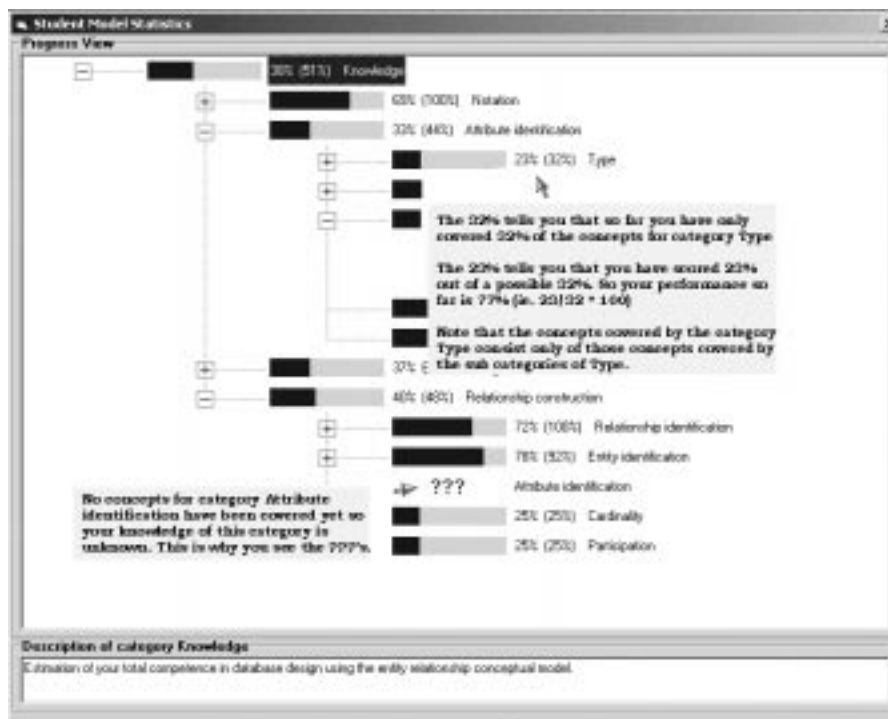


Fig. 4. A screenshot of the window showing the SM

6. Pre and Post Tests

The results of the pre/post tests are shown in Table 1. The maximum mark was 22 in both tests. Students on average performed equally well on the pre-test, providing evidence that both groups were of comparable abilities. In both groups, the post-test mean was significantly higher than the corresponding pre-test mean ($t=2.542$,

$p=0.015$ and $t=4.905$, $p=0.000$ respectively) revealing that students did improve their performance as the result of system interaction.

Table 1. Pre- and post-test scores

Group	No. of students	Pre-test mean (SD)	Post-test mean (SD)
Control	26	16.12 (1.82)	17.77 (1.45)
Experimental	40	16.23 (2.59)	17.13 (2.37)

Within each group students were divided into more and less able subgroups based on their performance in the pre-test. The more able group included students who scored above the mean in the pre-test. Table 2 shows the average gains (difference between the post- and pre-test marks) and standard deviations. A 2x2 two-way randomised ANOVA revealed that the system used had no significant effect on student gain. The mean gains between the more and less able students in both groups were found to be significantly different ($F(1,62)=45.881$, $p=0.001$), but there was no significant interaction between the student ability and the system used. Further analysis revealed that only the less able students in both groups achieved a significant improvement ($t=6.468$, $p=0.000$ and $t=3.534$, $p<0.005$ respectively). Hence, system interaction appears to be more beneficial to the less able students.

Table 2. The results on the pre- and post-tests for the more and less able students

Group	Students	Pre-test	Post-test	Gain
Control – less able	15	14.80 (1.01)	17.13 (1.25)	2.33 (1.40)
Control – more able	11	17.91 (0.83)	18.64 (1.29)	0.73 (1.74)
Exper. – less able	22	14.41 (1.89)	16.14 (2.36)	1.73 (2.29)
Exper. – more able	18	18.44 (1.20)	18.33 (1.78)	-0.11 (1.75)

7. Log Analysis

The average interaction times of the two groups were not significantly different, as well as the total number of problems attempted/completed. However, students in the experimental group abandoned more problems ($t=2.637$, $p=0.01$). The mean time per solved problem was the same, but the time per uncompleted problem is significantly higher for the experimental group ($t=1.817$, $p<0.1$). It is tempting to suggest that students in the e-KERMIT group were willing to spend more time attempting to solve the problems than those in the control group, but a more extensive evaluation is needed to make this claim. Table 3 shows these results.

The students, on average, accessed their full progress model 1.36 times. This is possibly an underestimate of the real mean, because students may have opened the progress window only once and kept it open in the background. The tutorial was accessed on average 0.53 times in addition to the first time when progress was assessed. Comparison of the more and less able students revealed no significant difference between the mean number of times the SM and tutorial were accessed. There is no real correlation between the number of times students accessed their progress and their gain on the post-test for both the more able and less able students

($r=0.344$ and $r=-0.146$ respectively). 51.2% of the experimental group students accessed the main progress window. 66.7% of the more able students opened the main progress window, while only 45.5% of the less able ones did the same.

Table 3. Statistics from the logs

	Control group	Experimental group
Interaction time (mins)	66.65 (21.35)	67.65 (27.4)
Time/uncompleted problem	16.50 (13.31)	20.55 (14.47)
Time/solved problem	13.48 (7.23)	13.97 (7.68)
Attempted problems	4.36 (1.40)	3.89 (2.57)
Abandoned problems	2.61 (1.40)	1.78 (1.25)
Solved problems	1.75 (1.14)	2.11 (1.39)

8. Subjective Analysis

The questionnaire first asked how much ER modelling experience the students had previously had. Valid responses were *only lectures*, *lectures plus some work* and *extensive*. No students in either group said they had had extensive experience; however a surprising number of students (62.5%) in the experimental group answered with lectures plus work, while only 20% of the students in the control group answered with this response. Despite the suggested difference in experience between the two groups, results from the pre-tests revealed that the groups were of equal competence.

The time taken to learn the interface was asked next. Possible responses were *less than 5 minutes*, *10 minutes*, *30 minutes* and *most of the session*. The mean times were 11.3 minutes and 14 minutes for the control and experimental respectively. The Mann-Whitney U test revealed this difference is statistically significant ($z=0.920$, $p=0.048$) which is understandable, because the interface of e-KERMIT is more complicated as it includes visualisation of the student model. Students were also asked to estimate how much they learnt and how much they enjoyed learning with the system, on a scale from 1 (lowest) and 5 (highest). There were no significant differences for the means for the amount learnt (3.2 and 3.1) and for enjoyment (3.4 and 3.6) for the experimental and the control group.

Students in the control group were asked whether they would like to have access to an open SM. Of the 31 students who completed the questionnaire, 1 student said no, 10 were unsure and 20 said yes, thus favouring an open student model approach.

Experimental group students were asked questions about the open SM. 27 students said they examined the summary panel in the main interface, 2 mostly viewed the progress window and 8 used both views equally¹. 9 students said they had difficulty in understanding their model, 18 had no problems and 9 said the model was understandable after explanation. Students were also asked whether they found the progress views useful, whether they examined their progress to identify weaknesses in their ER knowledge, and whether they felt that the opportunity to examine their progress assisted their learning. The responses to these questions are summarized in

¹ Not all students have answered all the questions.

Table 4. 68.97% of the students who examined their progress found the open SM to be useful. The progress model was used to help identify weaknesses in ER knowledge by majority of these students, thus supporting reflective activity.

Table 4. Responses to questions about the open student model (OSM)

Question	Yes	No	Do not know	N/A
OSM useful?	45%	7.5%	12.5%	22.5%
OSM used to identify weaknesses?	47.5%	40%	0%	0%
OSM assisted learning?	67.5%	5%	22.5%	0%

9. Conclusions and Further Work

Open student modelling is claimed to enhance learning because it promotes reflective activity in the student [5, 9, 19]. However, evaluation of the effects of open SM is currently lacking. This research has focused on how even a simple open SM can affect learning. For this purpose we enhanced an ITS for database design, to present an overview of the student's progress. The resulting system visualises the SM as a hierarchy of categories, which conveys the structure of the domain.

An evaluation study was conducted to compare students' learning with KERMIT and e-KERMIT. The study focused on how the open SM in e-KERMIT affected learning, whether students would inspect their models, and students' subjective opinion of their model. The subjective results from this study are quite encouraging. The majority of students who examined their model found it to be a useful tool to aid in learning. Although the study failed to demonstrate any statistically significant improvements in post-test scores, students who used e-KERMIT performed at least as well as those who used KERMIT. Given that students were using the systems for only one hour, of which fourteen minutes was used to learn the interface, any benefits of an open SM were not expected to have a significant effect on student performance.

Students were not told explicitly to explore the SM, yet around half of them did inspect it. Responses from the questionnaire indicate that students reflected on their progress and domain weaknesses to some extent, although the majority of students said that they consulted mostly the progress summary in the main interface.

This study supports other findings [13, 19] that system interaction benefits the less able students more than the more able ones. We found that although the performance of the more able students in both groups did not improve after system interaction, the amount that these students thought they learnt, on average, was 3.1, which was the same for the less able group, who, in fact, had improved.

This work is a first step in evaluating the effects of a simple open SM on learning. To estimate the real effects, it is necessary to perform a longer study, allowing the SM to become more sophisticated and giving students more time to learn about the domain. We plan to conduct such a study in 2002.

References

1. Aleven, V., Koedinger, K. Limitations of Student Control: Do Students Know When They Need Help? In: G. Gauthier, C. Frasson and K. VanLehn (eds): Proc. ITS'2000, Springer-Verlag (2000) 292-303

2. Brusilovsky, P., Schwarz, E., Weber, G. ELM-ART: an Intelligent Tutoring System on World Wide Web. Proc. ITS'96 (1996) 261-269
3. Bull, S. See Yourself Write: a Simple Student Model to Make Students Think. In: A. Jameson, C. Paris and C. Tasso (eds): Proc. UM'97, Springer (1997) 315-326
4. 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: J. Greer (ed), Proc. AIED'95 (1995) 501-508
5. Bull, S., Brna, P. & Pain, H. Extending the Scope of the Student Model. User Modeling and User Adapted Interaction, 5(1) (1995) 45-65
6. Bull, S., Brna, P. What does Susan Know that Paul Doesn't? (and vice versa): Contributing to each other's student model. In: B. du Boulay, R. Mizoguchi (eds), Proc. AIED 97, IOS Press, Amsterdam (1997) 568-570
7. Bull, S., Smith, M. A pair of student models to encourage collaboration. In: A. Jameson, C. Paris & C. Tasso (eds) Proc. 6th Int. Conf. on User Modeling, Springer (1997) 339-341
8. Bull, S., Brna, P. Enhancing peer interaction in the Solar system. In: P. Brna, M. Baker & K. Stenning (eds) Roles of communicative interaction in learning to model in Mathematics and Science: Proc. C-LEMMAS, Ajaccio, Corsica (1999)
9. Dimitrova, V., Self, J., Brna, P. Applying Interactive Open Learner Models to Learning Technical Terminology. In: M. Bauer, P. J. Gmytrasiewicz, J. Vassileva (eds.) Proc. UM 2001, Springer (2001) 148-157
10. Greer, J., McCalla, G., Collins, J., Kumar, V., Meagher, P., Vassileva, J. Supporting Peer Help and Collaboration in Distributed Workplace Environments, IJAIED, 9, (1998) 159-177.
11. Kay, J. The UM toolkit for Cooperative Student Modeling. User Modeling and User-Adapted Interaction, 4 (1995) 149-196
12. McCalla, G., Vassileva, J., Greer, J., Bull, S. Active Learner Modelling. In: G. Gauthier, C. Frasson, K. VanLehn (eds), Proc. ITS'2000, Springer (2000) 53-62
13. Mitrovic, A. Investigating students' self-assessment skills. In: M. Bauer, P.J.Gmytrasiewicz and J. Vassileva (eds) Proc. UM-2001, Springer (2001) 247-250
14. 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
15. Morales, R., Pain, H., Conlon, T. Effects of Inspecting Learner Models on Learners' Abilities. In: J. D. Moore, C. L. Redfield, W. L. Johnson (eds) Proc. AIED 2001 (2001) 434-445
16. 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.
17. Paiva, A., Self, J. TAGUS – a User and Learner Modelling Workbench. User Modeling and User-Adapted Interaction, 4 (1995) 197-226
18. Suraweera, P., Mitrovic, A. Designing an Intelligent Tutoring System for Database Modelling. In: M.J. Smith, G. Salvendy (eds) (2001) 745-749
19. White, B.Y., Shimoda, T. A, Frederiksen, J.R. Enabling Students to Construct Theories of Collaborative Inquiry and Reflective Learning: Computer Support for Metacognitive Development. Int. Journal on AI in Education, 10 (1999) 151-182
20. Zapata-Rivera, J.D., Greer, J.E. Inspecting and Visualizing Distributed Bayesian Student Models. In: G. Gauthier, C. Frasson, K. VanLehn (eds), ITS'2000, Springer (2000) 544-553
21. Zapata-Rivera, J-D. Supporting Negotiated Assessment Using Open Student Models. In: M. Bauer, P. J. Gmytrasiewicz, J. Vassileva (eds) UM 2001, Springer (2001) 295-297.