The Effect of Explaining on Learning: a Case Study with a Data Normalization Tutor

Antonija MITROVIC
Intelligent Computer Tutoring Group
Department of Computer Science and Software Engineering
University of Canterbury, New Zealand

Abstract: Several studies have shown that explaining actions increases students' knowledge. In this paper, we discuss how NORMIT supports self-explanation. NORMIT is a constraint-based tutor that teaches data normalization. We present the system first, and then discuss how it supports self-explanation. We hypothesized the self-explanation support in NORMIT would result in increased problem solving skills and better conceptual knowledge. An evaluation study of the system was performed, the results of which confirmed our hypothesis. Students who self-explained learnt constraints significantly faster, and acquired more domain knowledge.

1. Introduction

Although Intelligent Tutoring Systems (ITS) result in significant learning gains [9,11,12,13, 19], some empirical studies indicate that even in the most effective systems, some students acquire shallow knowledge. Examples include situations when the student can guess the correct answer, instead of using the domain theory to derive the solution. Aleven et al. [1] illustrate situations when students guess the sizes of angles based on their appearance. As the result, students have difficulties in transferring knowledge to novel situations, even though they obtain passing grades on tests.

The goal of ITSs is to enable students to acquire deep, robust knowledge, which they can use to solve different kinds of problems, and to develop effective meta-cognitive skills. Psychological studies [6,7] show that self-explanation is one of the most effective learning strategies. In self-explanation, the student solves a problem (or explains a solved problem) by specifying why a particular action is needed, how it contributes toward the solution of the problem, and what basic principles of the domain were used to perform the action.

This paper presents the support for self-explanation in NORMIT, a data normalization tutor. Section 2 reviews related work. Section 3 overviews the learning task, while the support for self-explanation is discussed in Section 4. The results of an evaluation study of NORMIT are presented in Section 5. The conclusions and avenues for future research are given in the final section.

2. Related Work

Metacognition includes processes involved with awareness of, reasoning and reflecting about, and controlling one's cognitive skills and processes. Metacognitive skills can be taught [5], and result in improved problem solving and better learning [1,8,18]. Of all metacognitive skills, self-explanation (SE) has attracted most interest within the ITS community. By explaining to themselves, students integrate new knowledge with existing knowledge. Furthermore, psychological studies show that self-explanation helps students to correct their misconceptions [7]. Although many students do not spontaneously self-explain, most will do so when prompted [8] and can learn to do it effectively [5].

SE-Coach [8] is a physics tutor that supports students while they study solved examples. The authors claim that self-explanation is better supported this way, than asking for explanation while solving problems, as the latter may put too big a burden on the student. In this system, students are prompted to explain a given solution for a problem. Different parts of the solution are covered with boxes, which disappear when the mouse is positioned over them. This masking mechanism allows the system to track how much time the student spends on each part of the solution. The system controls the process by modelling the self-explanation skills using a Bayesian network. If there is evidence that the student has not self-explained a particular part of the example, the system will require the student to specify why a certain step is correct and why it is useful for solving the current problem. Empirical studies performed show that this structured support is beneficial in early learning stages.

On the other hand, Aleven and Koedinger [1] explore how students explain their own solutions. In the PACT Geometry tutor, as students solve problems, they specify the reason for each action taken, by selecting a relevant theorem or a definition from a glossary. The performed evaluation study shows that such explanations improve students problem-solving and self-explanation skills and also result in transferable knowledge. In Geometry Explanation Tutor [2], students explain in natural language, and the system evaluates their explanations and provides feedback. The system contains a hierarchy of 149 explanation categories [3], which is a library of common explanations, including incorrect/incomplete ones. The system matches the student’s explanation to those in the library, and generates feedback, which helps the student to improve his/her explanation.

In a recent project [21], we looked at the effect of self-explanation in KERMIT, a database design tutor [19,20]. In contrast to the previous two systems, KERMIT teaches an open-ended task. In geometry and physics, domain knowledge is clearly defined, and it is possible to offer a glossary of terms and definitions to the student. Conceptual database design is a very different domain. As in other design tasks, there is no algorithm to use to derive the final solution. In KERMIT, we ask the student to self-explain only in the case their solution is erroneous. The system decides on which errors to initiate a self-explanation dialogue, and asks a series of question until the student gives the correct answer. The student may interrupt the dialogue at any time, and correct the solution. We have performed an experiment, the results of which show that students who self-explain acquire more conceptual knowledge than their peers [22].

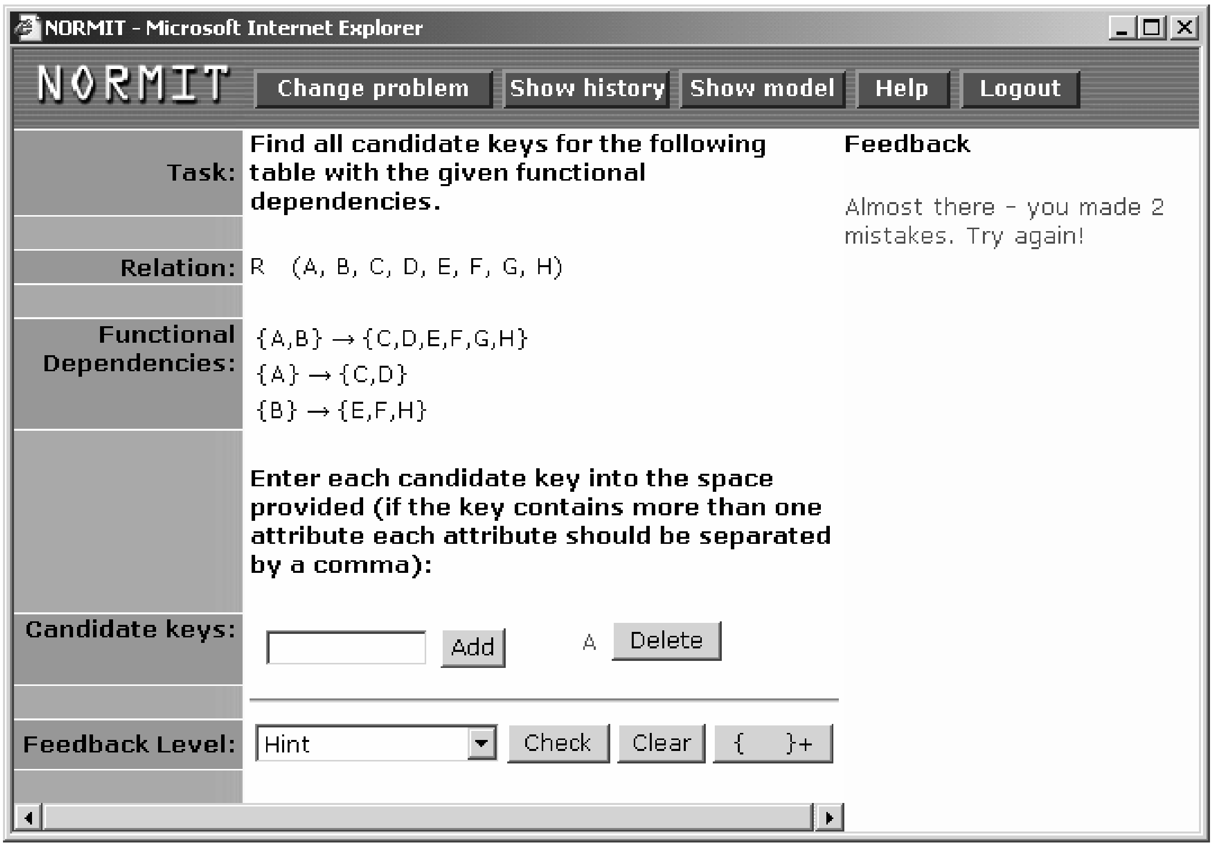
3. Learning Data Normalization in NORMIT

Database normalization is the process of refining a relational database schema in order to ensure that all tables are of high quality [10]. Normalization is usually taught in introductory database courses in a series of lectures that define all the necessary concepts, and later practised on paper by looking at specific databases and applying the definitions.

Like other constraint-based tutors [13,14,19], NORMIT is a problem-solving environment, which complements traditional classroom instruction. The emphasis is therefore on problem solving, not on providing information. Database normalization is a procedural task: the student goes through a number of steps to analyze the quality of a database. NORMIT requires the student to determine candidate keys (Figure 1), the closure of a set of attributes and prime attributes, simplify functional dependencies, determine normal forms, and, if necessary, decompose the table. The sequence is fixed: the student will only see a Web page corresponding to the current task. The student may submit a solution or request a new problem at any time. He/she may also review the history of the session, or examine the student model.

When the student submits the solution, the system analyses it and offers feedback. The first submission receives only a general feedback, specifying whether the solution is correct or not (as in Figure 1). If there are errors in the solution, the incorrect parts of the solution are shown in red. In Figure 1, for example, the student has specified A as the key of the given relation, which is incorrect. On the second submission, NORMIT provides a general description of the error, specifying what general domain principles have been violated. On the third submission, the system provides a more detailed message, by providing a hint as to how the student should change the solution. The student can also get a hint for every error. The correct solution is only available on request.

NORMIT is a Web-enabled tutor with a centralized architecture. As NORMIT is a constraint-based tutor [13,17], the domain model is represented as a set of 81 problem-independent constraints. For details of the system's architecture and implementation, please see [15].



Task: Find all candidate keys for the following table with the given functional dependencies.

Relation: R (A, B, C, D, E, F, G, H)

Functional Dependencies:
{A,B} → {C,D,E,F,G,H}
{A} → {C,D}
{B} → {E,F,H}

Enter each candidate key into the space provided (if the key contains more than one attribute each attribute should be separated by a comma):

Candidate keys: A

Feedback Level: Hint

Feedback

Almost there - you made 2 mistakes. Try again!

Fig. 1. A screenshot from NORMIT

4. Supporting Self-Explanation

NORMIT is a problem-solving environment, and therefore we ask students to self-explain while they solve problems. In contrast to other ITSs that support self-explanation, we do not expect students to self-explain every problem-solving step. Instead, NORMIT will require an explanation for each action that is performed for the first time. For the subsequent actions of the same type, explanation is required only if the action is performed incorrectly. We believe that this strategy will reduce the burden on more able students (by not asking them to provide the same explanation every time an action is performed correctly), and also that the system would provide enough situations for students to develop and improve their self-explanation skills.

Similar to the PACT Geometry Tutor and SE-Coach, NORMIT supports self-explanation by prompting the student to explain by selecting one of the offered options. In Figure 1, the student specified A as the candidate key incorrectly. NORMIT then asks the following question (the order in which the options are given is random, to minimize guessing):

This set of attributes is a candidate key because:

- *It is a minimal set of attributes*
- *Every value is unique*
- *It is a minimal set of attributes that determine all attributes in the table*
- *It determines the values of all other attributes*
- *All attributes are keys*
- *Its closure contains all attributes of the table*

The candidate answers to choose from are not strict definitions from the textbook, and the student needs to reason about them to select the correct one for the particular state of the problem. For this reason, we believe that the support for self-explanation in NORMIT (i.e. explanation selection) is adequate support. In this way, self-explanation is not reduced to recognition, but truly requires the student to re-examine his/her domain knowledge in order to answer the question. Therefore, this kind of self-explanation support requires recall and is comparable to generating explanations. Furthermore, this kind of self-explanation support is easier to implement in comparison to explaining in a natural language. Although it may seem that explaining in a natural language would give better results than selecting from pre-specified options, Aleven, Koedinger and Popescu [4] show that this is not necessarily the case: in their study there was no significant difference between students who explained by selecting from menus, and students who explained in English.

If the student's explanation is incorrect, he/she will be given another question, asking to define the underlying domain concept (i.e. candidate keys). For the same situation, the student will get the following question after giving an incorrect reason for specifying attribute A as the candidate key:

A candidate key is:

- *an attribute with unique values*
- *an attribute or a set of attributes that determines the values of all other attributes*
- *a minimal set of attributes that determine all other attributes in the table*
- *a set of attributes the closure of which contains all attributes of the table*
- *a minimal superkey*
- *a superkey*
- *a key other than the primary key*
- *A candidate key is an attribute or a set of attributes that determine all other attributes in the table and is minimal. The second condition means that it is not possible to remove any attributes from the set, and still have the remaining attributes to determine the other attributes in the table.*

In contrast to the first question, which was problem-specific, the second question is general. If the student selects the correct option, he/she will resume with problem solving. In the opposite case, NORMIT will provide the correct definition of the concept.

In addition to the model of the student's knowledge, NORMIT also models the student's self-explanation skills. For each constraint, the student model contains information about the student's explanations related to that constraint. The student model also stores the history of student's explanation of each domain concept.

5. Experiment

We performed an evaluation study with the students enrolled in an introductory database course at the University of Canterbury. Our hypothesis was that self-explanation would have positive effects on both procedural knowledge (i.e. problem solving skills) and conceptual knowledge. Prior to the experiment, the students had four lectures and one tutorial on data normalization. The system was demonstrated in a lecture on October 5, 2004 (during the last week of the course), and was open to the students a day later. The students in the control group used the basic version of the system, while the experimental group used NORMIT-SE, the version of the system that supports self-explanation. The participation was voluntary, and 61 out of 124 students enrolled in the course used the system. The students were free to use NORMIT when and for how long they wanted.

The pre-test (with the maximum mark of 4) was administered on-line at the beginning of the first session. We developed two tests, each having four multichoice questions. The first two questions required students to identify the correct solution for a given problem, while for the other two students needed to identify the correct definition of a given concept. These two tests were randomly used as the pre-test. The post-test was administered as a part of the final examination on October 29, 2004.

Table 1. Mean system interaction details (standard deviations given in parentheses)

	NORMIT	**NORMIT-SE**
Students	27	22
Sessions	2.9 (1.95)	2.4 (1.7)
Time spent (min.)	231 (202)	188 (167)
Attempted problems	**16.7 (11.2)**	**11.9 (10.4)**
Completed problems (%)	81.9 (22.5)	80.4 (16.2)
Pre-test (%)	55.6 (26.2)	64.77 (26.3)
Post-test (%)	51.3 (15.4)	53.61 (22.3)

We collected data about each session, including the type and timing of each action performed by the student, and the feedback obtained from NORMIT. Twelve students have logged on to the system for a very short time, and have solved no problems, and we excluded their logs from analyses. Table 1 reports some statistics about the remaining students. The average mark on the pre-test for all students was 59.7% (sd = 26.4). The groups are comparable, as there is no significant difference on the pre-test.

There was no significant difference between the two groups on the number of sessions or the total time spent with the system. The number of attempted problems ranged from 1 to 49 (the total number of problems in the system is 50). The difference between the mean number of attempted problems for the two groups is significant (p=0.067). We believe this is due to more time needed for self-explanation for the experimental group students. Both groups of students were equally successful at solving problems, as there was no significant difference on the percentage of solved problems.

As explained earlier, the post-test was administered as a part of the final examination for the course. We decided to measure performance this way because the study was not controlled, and this was the only way to ensure that each participant sits the post-test. However, this decision also dictated the kinds of questions appearing in the post-test. As the consequence, our pre- and post-tests are not directly comparable. The post-test was longer, with a maximum of 29 marks. Therefore we cannot compare the students' performance before and after the study.

There was no significant difference between the post-test results of the two groups. However, it is important to note that 60% of the control group students and 73% of the experimental group students logged on to NORMIT for the first time just a day or two before the post-test. Furthermore, the students on average spent only 3-4 hours working with the system. Therefore, it is not reasonable to expect significant difference after such short interaction times.

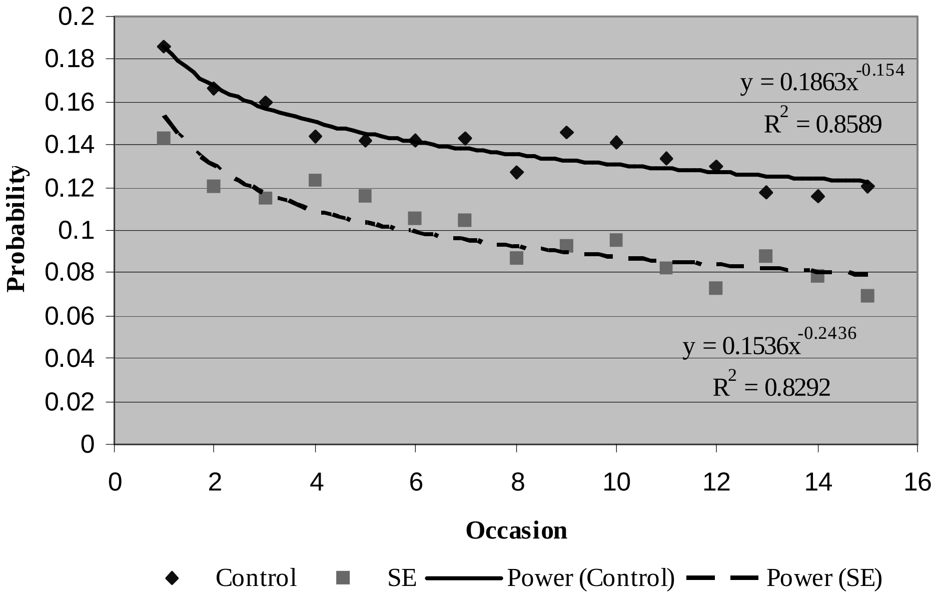
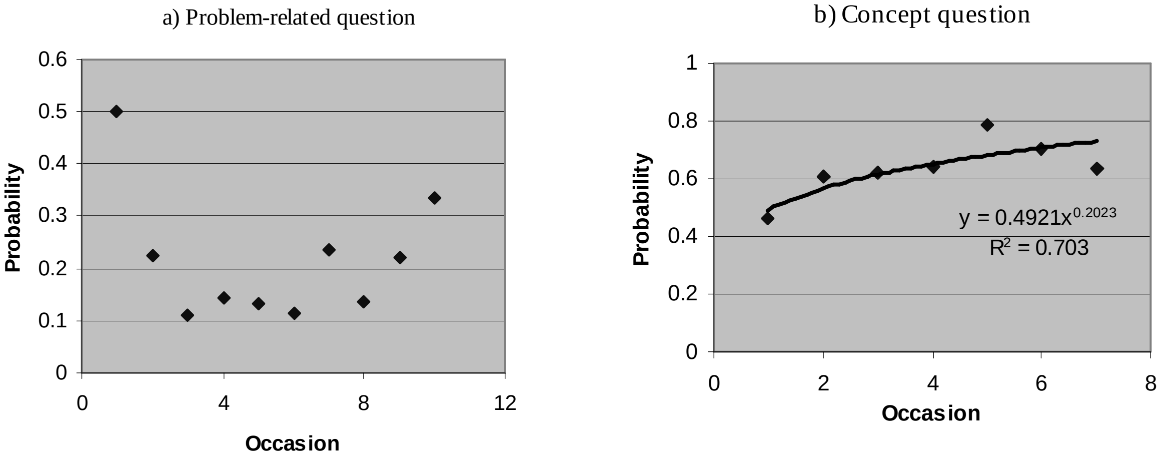


Fig. 2. Learning constraints

Figure 2 shows how students learnt constraints. We looked at the proportion of violated constraints following the n^{th} occasion when a constraint was relevant, averaged across all students and all constraints. The R^2 fits to the power curves are good for both groups, showing that all students learnt constraints by using the system. The learning curve for the experimental group shows that these students are less likely to violate constraints and learn constraints faster than their peers. The learning rate of the experimental group (.24) is higher than the learning rate of the control group (.15). We have also analysed individual learning curves, for each participant in the study. The learning rates of students in the experimental group are significantly higher than those of the control group students (p=0.014). This finding confirms our hypothesis that self-explanation has a positive effective on students' domain knowledge.

We also analysed the data about students' self-explanations. There were 713 situations where students were asked to self-explain. On average, a student was asked 32.4 problem-oriented SE questions (i.e. the first question asked when a student makes a mistake), and 23.2 concept-oriented SE questions, and correct explanations were given in 31.9% and 56.7% of the cases respectively. Figure 3.a shows the probability of giving a correct answer to the problem-related SE question averaged over all occasions and all participants. As can be seen, this probability varies over occasions, but always stays quite low. Therefore, students find it hard to give reasons for their actions in the context of the current problem. Some concepts are much more difficult for students to learn than others. For example, out of the total of 132 situations when students who were asked to explain why a set of attributes is a candidate key, the correct answer was given in only 23 cases. Figure 3.b shows the same probability for the question asking to define a domain concept (conceptual question). As the figure illustrates, the students were much better at giving definitions of domain concepts. In the case of candidate keys, although students were pretty bad in justifying their choice of candidate key in a particular situation (when the correct answer was given in 17.4% of the cases), when asked to define a candidate key, they were correct in 45% of the cases. Figure 3.b shows a regular increase of the probability of correct explanation, showing that the students did improve their conceptual knowledge through explaining their actions.


a) Problem-related question

b) Concept question

Fig. 3. Student's performance on self-explanation

6. Conclusions

Self-explanation is known to be an effective learning strategy. Since ITSs aim to support good learning practices, it is not surprising that researches have started providing support for self-explanation. In this paper, we present NORMIT-SE, a data normalization tutor, and describe how it supports self-explanation. NORMIT-SE is a problem-solving environment, and students are asked to explain their actions while solving problems. The student must explain every action that is performed for the first time. However, we do not require the student to explain every action, as that would put too much of a burden on the student and reduce motivation. NORMIT-SE requires explanations in cases of erroneous actions. The student is asked to specify the reason for the action, and, if the reason is incorrect, to define the domain concept that is related to the current task. If the student is not able to identify the correct definition from a menu, the system provides the definition of the concept.

We performed a pilot study of the system in a real course in 2002 [16]. In 2003 we performed an evaluation study, but did not have enough participants to draw any conclusions. This paper presented a study performed in 2004, which had more participants than the previous two. The results of the study support our hypothesis: students who self-explained learned constraints significantly faster than their peers who were not asked to self-explain. There was no significant difference between the two conditions on the post-test performance, and we believe that is due to the short times the participants spent interacting with the system. Furthermore, the analysis of the self-explanation behaviour shows that students find problem-specific question (i.e. explaining their action in the context of the current problem state) more difficult than defining the underlying domain concepts. The students' conceptual knowledge improved regularly during their interaction with NORMIT-SE.

There are two main avenues for future work. At the moment, the student model in NORMIT contains a lot of information about the student's self-explanation skills that is not used. We plan to use this information to identify domain concepts for which the student needs more instruction. Furthermore, the self-explanation support itself may be made adaptive, so that different support would be offered to students who are poor self-explainers in contrast to students who are good at it.

Acknowledgements: We thank Li Chen and Melinda Marshall for implementing NORMIT's interface.

References

1. Aleven, V., Koedinger, K., Cross, K. Tutoring Answer Explanation Fosters Learning with Understanding. In *Proc. Int. Conf. Artificial Intelligence and Education*, 1999, pp. 199-206.
2. Aleven, V., Popescu, O., Koedinger, K. Towards Tutorial Dialogue to Support Self-Explanation: Adding Natural Language Understanding to a Cognitive Tutor. Int. J. Artificial Intelligence in Education, vol. 12, 2001, 246-255.
3. Aleven, V., Popescu, O., Koedinger, K. Pilot-Testing a Tutorial Dialogue System that Supports Self-Explanation. In *Proc. Int. Conf. Intelligent Tutoring Systems*, Biarritz, France, 2002, pp. 344-354.
4. Aleven, V., Popescu, O., Koedinger, K. A Tutorial Dialogue System to Supports Self-Explanation: Evaluation and Open Questions. In U. Hoppe, F. Verdejo and J. Kay (eds) *Proc. Int. Conf. Artificial Intelligence in Education*, Sydney, 2003, pp. 39-46.
5. Bielaczyc, K., Pirolli, P., Brown, A. Training in Self-Explanation and Self-Regulation Strategies: Investigating the Effects of Knowledge Acquisition Activities on Problem-solving. Cognition and Instruction, vol. 13, no. 2, 1993, 221-252.
6. Chi, M. Self-explaining Expository Texts: The dual processes of generating inferences and repairing mental models. Advances in Instructional Psychology, 2000, 161-238.
7. Chi, M. Bassok, M., Lewis, W., Reimann, P., Glaser, R. Self-Explanations: How Students Study and Use Examples in Learning to Solve Problems. Cognitive Science, vol. 13, 1989, 145-182.
8. Conati, C., VanLehn, K. Toward Computer-Based Support of Meta-Cognitive Skills: a Computational Framework to Coach Self-Explanation. Int. J. Artificial Intelligence in Education, vol. 11, 2000, 389-415.
9. Corbett, A., Trask, H., Scarpinatto, K., Handley, W. A formative evaluation of the PACT Algebra II Tutor : support for simple hierarchical reasoning. In *Proc. Int. Conf. Intelligent Tutoring Systems*, San Antonio, 1998, pp. 374-383.
10. Elmasri, R., Navathe, S. B. *Fundamentals of database systems*. Benjamin/Cummings, 2003.
11. Gertner, A.S, VanLehn, K. ANDES: A Coached Problem-Solving Environment for Physics. In G. Gauthier, C. Frasson, and K. VanLehn, (eds.), *Proc. Int. Conf. ITS*, Montreal, 2000, pp. 133-142.
12. Grasser, A., Wiemer-Hastings, P., Kreuz, R. AUTOTUTOR: A Simulation of a Human Tutor. Journal of Cognitive Systems Research, vol. 1, no. 1, 1999, 35-51.
13. Mitrovic, A., Ohlsson, S. Evaluation of a constraint-based tutor for a database language. Int. J. Artificial Intelligence in Education, vol. 10, no. 3-4, 1999, 238-256.
14. Mitrovic, A., Suraweera, P., Martin, B, Weerasinghe, A. DB-suite: Experiences with Three Intelligent, Web-based Database Tutors. Interactive Learning Research, vol. 15, no. 4, 409-432.
15. Mitrovic, A. NORMIT, a Web-enabled tutor for database normalization. In Proc. Int. Conf. Computers in Education, Auckland, New Zealand, 2002, pp. 1276-1280.
16. Mitrovic, A. Supporting Self-Explanation in a Data Normalization Tutor. In: V. Aleven, U. Hoppe, J. Kay, R. Mizoguchi, H. Pain, F. Verdejo, K. Yacef (eds) Supplementary proceedings, AIED 2003, 2003, pp. 565-577.
17. Ohlsson, S. Constraint-based Student Modeling. In Student Modeling: the Key to Individualized Knowledge-based Instruction. 1994, 167-189.
18. Schworm, S., Renkl, A. Learning by solved example problems: Instructional explanations reduce self-explanation activity. Proc. 24th Cognitive Science Conf., 2002, pp. 816-821.
19. Suraweera, P., Mitrovic, A. An Intelligent Tutoring System for Entity Relationship Modelling. Int. J. Artificial Intelligent in Education, vol. 14, no. 3-4, 2004, 375-417.
20. Suraweera, P., Mitrovic, A. KERMIT: a Constraint-based Tutor for Database Modeling. In *Proc .Int. Conf. Intelligent Tutoring Systems*, Biarritz, France, 2002, pp. 377-387.
21. Weerasinghe, A., Mitrovic, A. Enhancing learning through self-explanation. *Proc. Int. Conf. Computers in Education*, Auckland, New Zealand, 2002, pp. 244-248.
22. Weerasinghe, A., Mitrovic, A. Supporting Self-Explanation in an Open-ended Domain. In: M. Gh. Negoita, R. J. Howlett and L. C. Jain (eds) Proc. 8th Int. Conf. Knowledge-Based Intelligent Information and Engineering Systems KES 2004, Berlin, Springer LNAI 3213, 2004, pp. 306-313.