Effects of Self-Explanation in an Open-ended Domain

Amali WEERASINGHE and Antonija MITROVIC
Intelligent Computer Tutoring Group
Department of Computer Science, University of Canterbury
Private Bag 4800, Christchurch, New Zealand
{acw51, tanja}@cosc.canterbury.ac.nz

Abstract: We are interested in investigating whether self-explanation can be used in an open-ended domain. For this purpose, we enhanced KERMIT, an intelligent tutoring system that teaches conceptual database design. The resulting system, KERMIT-SE, supports self-explanation by engaging students in tutorial dialogues when their solutions are erroneous. The results of the evaluation study indicate that self-explanation leads to improved conceptual and procedural knowledge.

1. Introduction

Empirical studies indicate that students acquire shallow knowledge even in the most effective Intelligent Tutoring Systems (ITS) [1]. Self-explanation has been shown to facilitate the acquisition of deep knowledge [2]. However, only two ITSs support self-explanation. SE-Coach [3] prompts students to explain solved physics examples. In the PACT Geometry Tutor [1] students explain solution steps by selecting definitions and theorems from a glossary, which results in improved problem solving skills. Problem solving activities in these two domains are well structured, and self-explanation expected from students can be clearly defined. However, it is challenging to incorporate self-explanation in an open-ended task. Database design is an example of such tasks: the final result can be defined in abstract terms, but there is no algorithm to find it.

In this paper we present the self-explanation support added to KERMIT [5], an ITS in which students practice database design using the Entity Relationship (ER) data model. 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 [4] to model the domain and student's knowledge. In this paper, we present KERMIT-SE [6], the enhancement that facilitates self-explanation. The results of the evaluation study are presented in Section 3. The conclusions and directions for future research are given in the final section.

2. Design and Implementation of KERMIT-SE

KERMIT-SE prompts for self-explanation only when the student violates a constraint. We have analysed different students' errors and arranged them into a hierarchy, ordered from basic domain principles to more complicated ones. Violated constraints for each type of error are represented as leaves of the hierarchy. We designed a tutorial dialogue for each type of error. Dialogues dealing with basic errors contain only a simple explanation of the error. An example of such a dialogue is "*You have connected an entity A to entity B directly. Entities cannot be directly connected to each other*." Other dialogues are aimed to assist students to understand complex domain concepts. One of the longest dialogues consists of seven questions.

When the student submits a solution, the student modeller evaluates it against the constraint base and identifies violated constraints. The pedagogical module then searches for the first tutorial dialogue for the violated constraints, and displays the error. In addition, a list of general feedback messages for each violated constraint is shown. Students can initiate self-explanation using the *More Help* button. Figure 1 presents a sample dialogue, which may occur between the student and the tutor. Initially, the student is informed of the error, and is asked to specify his/her modeling decision (*tutor-1*). A list of possible answers is provided from which the correct one can be selected. If the student fails to provide the correct answer or asks for help (*student-1*), s/he will be asked a more specific question (*tutor-2*). However, if s/he fails to correct the mistake after going through a series of questions, the tutor will specify the correct action to be performed together with a brief description about the fundamental principle that needs to be learnt (*tutor-7, tutor-8*). KERMIT-SE uses simple questions (*tutor-1*), fill-in-a-blank (*tutor-8*), or true-false questions. Since all dialogues point out the error (*tutor-1*), students can reflect on their problem solving procedure, which is another important meta-cognitive skill.

Tutor-1: You've modeled *TOPIC* as an entity. What is the general rule to decide whether something is an entity?
Student-1*: I don't know*
Tutor-2: Let's try a simple thing. How do we model characteristics of an entity?
Student-2*: Using Attributes*
Tutor-3: Great! Now tell me why modeling *Topic* as an entity contradicts with the problem description?
Student-3*: I'm not sure*
Tutor-4: Don't worry. I'm here to help you. Check carefully whether you can find any attributes for *Topic*. {After sometime} Please tell me whether you managed to find any attributes.
Student-4*: No*
Tutor-5: Fine. It doesn't have any attributes. So now can you tell me why modeling *Topic* as an entity contradicts with the problem description?
Student -5*: Because it doesn't have any attributes to describe its characteristics.*
Tutor-6: Great! Now tell me the best way to model *Topic*?
Student-6*: I need more help*
Tutor-7: Ok. If *Topic* has no attributes, it cannot be an entity. So it should either be a relationship or an attribute. So what is the best way to model *Topic?*
Student-7*: Attribute*
Tutor-8: Great! Now use the correct word to fill in the blanks so that you'll remember what you learnt.
…………….. are used to describe characteristics of an ………….. in a ER model
Student-8*: attributes, entity*
Tutor-9: Excellent, Go ahead and make the changes. Submit the solution when you're done.

Fig. 1. A sample tutorial dialogue

3. Evaluation Study

An evaluation study was conducted in July 2002 with students enrolled in an introductory database course at the University of Canterbury. The experimental group used KERMIT-SE

(53 students), while the control group (72 students) used KERMIT. The experiment was carried out during normal lab hours over the duration of two weeks. Only 19 the experimental group students self-explained (they had control over that via the *More Help* button). We are interested in these students (*self-explainers* in Table 1), as the rest of the group has not self-explained (*non self-explainers*). These two subgroups are comparable, as there is no significant difference on the pre-test.

Table 1. Mean system interaction details

	Self-explainers	Non self-explainers
Time (min.)	133:21(30:44)	79:13 (47:41)
Attempted problems	8.21 (2.42)	5.29 (3.17)
Completed problems	6.36 (2.31)	3.65 (2.29)
No. of post-tests	18	17
Pre-test	79.32 (13.16)	73.17 (20.47)
Post-test	79.76 (12.22)	77.37(16.76)

Self-explainers spent significantly more time on problem solving ($t=5.01$, $p<0.001$) than non self-explainers, and also attempted and solved significantly more problems. The self-explainers went through 6.95 dialogues, with an average of 57.61% of correct responses. We analysed the answers to the first question, which prompts students to explain domain concepts (Fig.2). The probabilities of correct answers on the first and subsequent occasions were averaged over all error types and all students. The fit to the power curve is very good, indicating that students do learn by explaining.

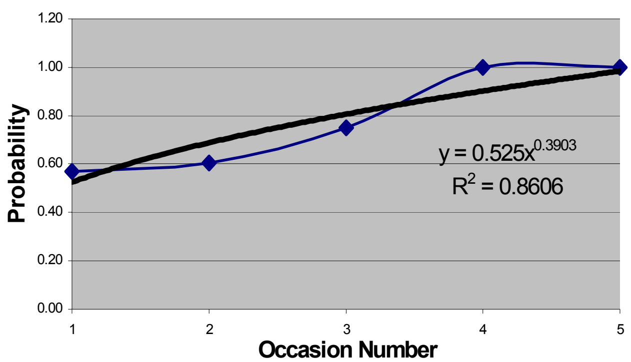

Fig. 2. Performance on the first question in the dialogues

4. Conclusions

This research focuses on incorporating self-explanation into a tutor that teaches ER modeling. KERMIT-SE supports self-explanation by engaging students in tutorial dialogues about errors they make. Students are asked problem-specific and general questions, and can select answers from menus. An evaluation study was conducted in July 2002 to investigate whether guided self-explanation would improve students' learning in the domain of database modeling. The results indicate that self-explanation leads to improvement in problem solving and in answering questions about domain knowledge. We plan to enhance the student model in KERMIT-SE to provide adaptive self-explanation. i.e. to provide support self-explain based on the student's existing self-explanation skill.

References

1. Aleven, V., Koedinger, K. R., Cross, K. Tutoring Answer Explanation Fosters Learning with Understanding. In: Artificial Intelligence in Education, Lajoie, S.P. and Vivet, M.(eds.), IOS Press (1999) 199-206.
2. Chi, M. T. H. Self-explaining Expository Texts: The dual processes of generating inferences and repairing mental models. Advances in Instructional Psychology, (2000) 161-238.
3. 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, 11 (2000) 389-415.
4. Ohlsson, S. Constraint-based Student Modeling. In: Greer, J.E., McCalla, G (eds) Proc. of Student Modeling: the Key to Individualized Knowledge-based Instruction, Springer-Verlag Berlin (1994) 167-189.
5. Suraweera, P., Mitrovic, A. KERMIT: a constraint-based tutor for database modeling. In: S. Cerri, G. Gouarderes and F. Paraguacu (eds.) Proc. ITS'2002, Biarritz, France, LCNS 2363, 2002, 377-387.
6. Weerasinghe, A., Mitrovic, A. (2002) Enhancing learning through self-explanation. In: Kinshuk, R. Lewis, K. Akahori, R. Kemp, T. Okamoto, L. Henderson, C-H Lee (eds.) Proc. ICCE 2002, 244-248.