Assessing the Impact of Positive Feedback in Constraint-based Tutors

Devon Barrow[1], Antonija Mitrovic[1],
Stellan Ohlsson[2] and Michael Grimley[1]

[1]University of Canterbury, Private Bag 4800
Christchurch 8140, New Zealand
dba46@student.canterbury.ac.nz
{tanja.mitrovic,michael.grimley@canterbury.ac.nz}

[2]Department of Psychology, University of Illinois, Chicago
stellan@uic.edu

Abstract. Most existing Intelligent Tutoring Systems (ITSs) are built around cognitive learning theories, such as Ohlsson's theory of learning from performance errors and Anderson's ACT theories of skill acquisition, which focus primarily on providing negative feedback, facilitating learning by correcting errors. Research into the behavior of expert tutors suggest that experienced tutors use positive feedback quite extensively and successfully. This paper investigates positive feedback; learning by capturing and responding to correct behavior, supported by cognitive learning theories. Our aim is to develop and implement a systematic approach to delivering positive feedback in ITSs. We report on an evaluation study done in the context of SQL-Tutor, in which the control group used the original version of the system giving only negative feedback, while the experimental group received both negative and positive feedback. Results show that the experimental group students needed significantly less time to solve the same number of problems, in fewer attempts compared to those in the control group. Students in the experimental group also learn approximately the same number of concepts as students in the control group, but in much less time. This indicates that positive feedback facilitates learning and improves the effectiveness of learning in ITSs.

1 Introduction

Intelligent Tutoring Systems (ITSs) continue to grow in popularity and application. The use of computers to support education and learning has more than doubled between 1984 (36.2%) and 1997 (84%) [3]. ITSs are at the forefront of these technologies. There is however still considerable work to be done in making such systems more effective. Current ITSs provide one-on-one tutoring at relatively low

cost and the added flexibility with regard timing, location and amount of the tutoring experience. Evaluation of a LISP tutor showed that students in the experimental group completed problems in one third the time of those under control conditions with improvement in learning of 1 standard deviation [4]. ANDES, a tutoring system for teaching Physics, improves performance by 0.9 standard deviations [5], while SQL-Tutor improves performance by 0.65 standard deviations in just two hours of interaction with the system [6].

However, more research is still needed in order to develop a theory of tutoring which would link the various learning mechanisms to the types of information the learner needs and hence what the tutor (ITS) should provide. Current hypotheses propose that student learning is correlated with the tutor's use of knowledge-construction activities e.g. dynamic plan scaffolding, and refection/generalization techniques [5, 7]. There is a move to incorporate these techniques into ITSs through modeling of tutorial actions and strategies as observed with expert human tutoring [7]. Positive feedback is one of the teaching strategies which continue to surface throughout tutoring protocols. Work being done with tutoring protocols at the University of Illinois at Chicago [8] shows extensive and effective use of positive feedback by experienced human tutors. A logical step for ITSs therefore seems to be extending the proposed model of tutoring to incorporate positive feedback.

In this paper, we describe a study investigating the effect of positive feedback performed in the context of SQL-Tutor. We start by briefly describing this ITS in Section 2, and then describe our approach to providing positive feedback in Section 3. Section 4 presents the study and the results obtained, while the conclusions are given in the final section.

2 SQL-Tutor

SQL-Tutor assists university-level students in acquiring the knowledge and skills necessary to create SQL queries. SQL is the dominant database query language, which students find hard to learn. SQL-Tutor is a mature ITS, designed as a practice environment with the prerequisite that students be previously exposed to the SQL concepts in lectures. In the web-enabled version of SQL-Tutor, the architecture of which is shown in Figure 1, each student is assigned a unique web session. Students submit solutions which are sent to the student modeller for analysis. The student modeller identifies any errors or mistakes and updates the student model accordingly to reflect student progress within the domain. For more details of SQL-Tutor, please see [6].

To check the correctness of the student's solution, SQL-Tutor compares it to the correct solution, using domain knowledge, represented as a set of 700 constraints. A constraint consists of a relevance condition C_r, which checks whether the constraint is appropriate for a particular student's solution, and a satisfaction condition, C_s. A solution is correct if it satisfies the satisfaction conditions of all relevant constraints.

Once the student's solution is evaluated, the student model passes information to the pedagogical module which generates the appropriate feedback. If any constraints are violated, SQL-tutor will provide feedback on them. In the case where the solution

is correct or the student requires a new problem to work on, the pedagogical module uses the information from the student model to select an appropriate problem.

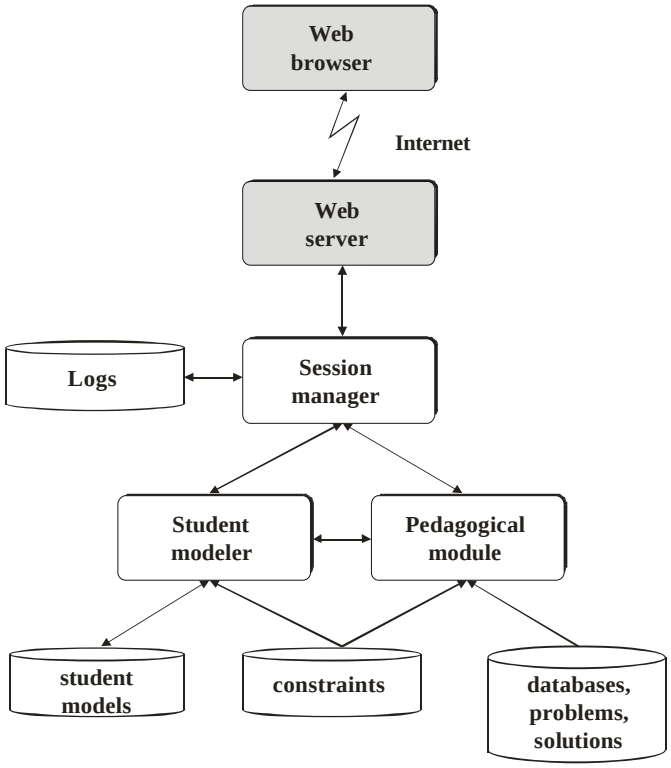

Fig. 1. Architecture of SQL-Tutor

SQL-Tutor provides feedback on demand only, when the student submits the solution. The system offers six levels of feedback, differing in the amount of detail provided to the student. On the first attempt, the system only informs the student whether the solution is correct or not. All other feedback levels provide negative feedback, i.e. feedback on errors. The second level (*Error Flag*) points the part of the solution that is incorrect. The third level (*Hint*) provides a description of one error, pointing out where exactly the error is, what constitutes the error (performing blame allocation) and referring the student to the underlying domain principle that is violated (revising student's knowledge). The hint message comes directly from the violated constraint. The automatic progression of feedback levels ends at the hint level; to obtain higher levels of feedback, the student needs to explicitly require them. For example, the student can ask for the hint message for all violated constraints (All Errors), a partial solution (showing the correct version of one part of the solution that is wrong), or a complete solution for the problem.

3 Providing Positive Feedback

Consider a situation when a student is attempting a problem but is not entirely sure of what to do. That student makes a tentative attempt at the next step and surprisingly it turns out to be correct. The student is likely to store that piece of 'correct' knowledge if they are aware that their action is indeed correct. Therefore there must be some feedback mechanism which confirms to the student that their action is correct. Experienced human tutors support the learning process by providing feedback to confirm the correctness of the student's action. This is only one of the many situations where positive feedback is used quite effectively to support student learning. It therefore appears that one of the ways in which positive feedback works is by reducing the number of tentative steps made by a student, creating and storing new knowledge chunks in cases where the student was lacking, or strengthening in situations where there was uncertainty and doubt.

Many student steps are tentative; the student is guessing, or at least rather uncertain as to what to do. In either case, if such a move happens to be correct then providing assurance to the student of its accuracy helps to reduce uncertainty and apprehension. We hypothesize that positive feedback works by helping to reduce the amount of uncertainty associated with student actions and reinforce existing knowledge; in some cases creating new knowledge and by so doing reduces both the time taken by students to solve problems and the number of errors made.

We developed a systematic approach to delivering positive feedback. This approach addresses three main issues: timing, content and presentation of positive feedback messages. Regarding timing of positive feedback, we considered two questions: when and how often should positive feedback be given. We proposed that positive feedback be given when the student submits a solution, as is currently the case with negative feedback in SQL-Tutor. However, the system should not give positive feedback on each correctly used constraint, as the amount of such feedback would be overwhelming. Instead, we identified events which add positive feedback only on selected constraints to the other messages given to the student. The positive feedback provided to a particular student will depend not only on the submitted solution, but also on the student's knowledge (as captured by the student model) and the state of interaction. We developed four general cases when positive feedback should be given:

a) *When the student is expressing uncertainty but nevertheless does the right* thing. This situation occurs when the student has previously made errors in a similar situation, but has made a correct attempt at the current problem. In this case, positive feedback reinforces the newly learnt domain principles.
b) *When the student is too paralyzed to do anything at all*. In such situations the student is unable to proceed without additional aid. The student can only move forward if the tutor provides a direct hint on what the solution should be. This hint can be requested by the student, or provided automatically by the system after a period of inactivity. If the student uses the hint successfully, then positive feedback should be given. Based on our hypothesis, positive feedback should increase student confidence, decrease uncertainty over the student's existing knowledge, decrease uncertainty over the knowledge given within the hint

statement which was once lacking and strengthen the connection between these two pieces of knowledge chunks.

c) *When the student has overcome aspects of the domain commonly agreed upon as being difficult and challenging.* Certain domain principles are difficult and challenging increasing cognitive load and the amount of active information processing required from the student. We propose giving positive feedback when the student correctly and adequately deals with such situations. This requires cognitive task analysis to identify such domain principles.

d) *When the problem or task has been successfully completed and at other major goals within the tutoring session.* Solving a problem represents a significant achievement. If a student gets a very difficult problem correct on the first attempt or satisfies a difficult constraint the first time it is relevant, then it potentially signifies that the student has mastered those aspects of the domain. In this case we give positive feedback to indicate to the student the magnitude of their accomplishment and to reinforce that correct response.

Based on these cases, we developed a set of rules for providing positive feedback. Rule 1 generates positive feedback for each constraint satisfied for the first time (i.e. all previous attempts on that constraint were incorrect). Rule 2 provides positive feedback when the student was given a hint on how to solve the problem by SQL-Tutor after a period of inactivity, and has managed to apply the hint successfully (i.e. the student satisfied the constraint for the first time in its history). In that case, the

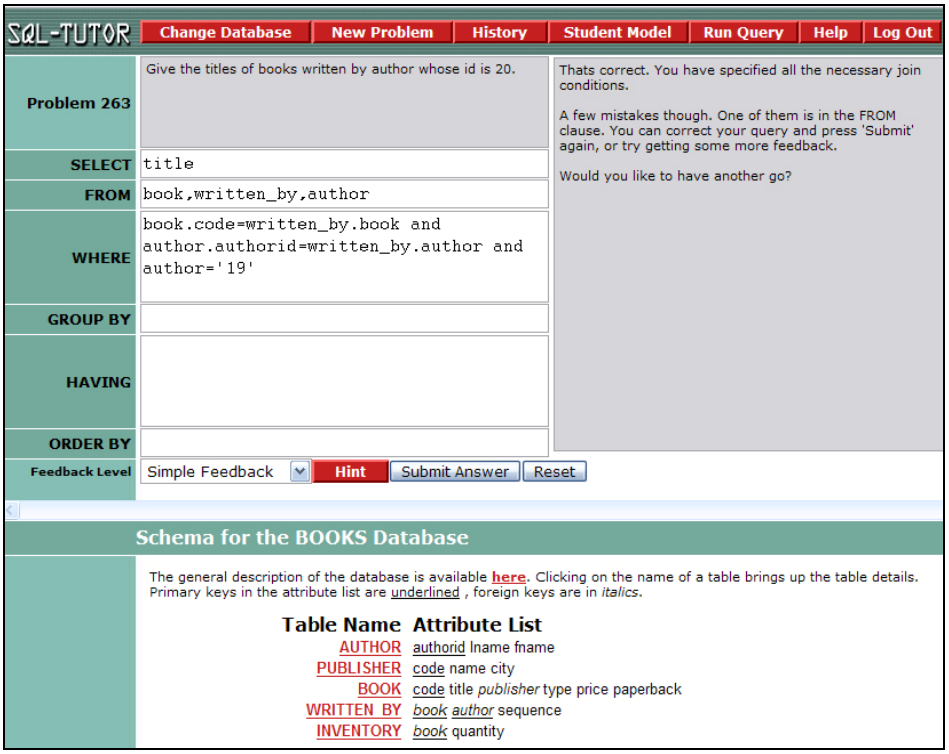

Fig. 2. Screenshot of SQL-Tutor providing both positive and negative feedback

student will be given positive feedback on the newly satisfied constraint. Rule 3 is similar to the previous rule, but covers the situation when the student requested a hint, rather than being provided with one automatically. Rule 4 provides positive feedback on a constraint that a student used correctly earlier in the session, but then kept violating it. When the student uses that constraint correctly again, rule 4 would generate reinforcing positive feedback on that constraint. Rule 5 covers a situation when a student was given a hint, and has consequently satisfied a constraint which was previously violated. Rule 6 covers a similar situation, but differs from rule 5 in the fact that the student requested a hint. Rule 7 generates positive feedback in the case of a difficult problem being solved correctly, while rule 8 provides positive feedback after satisfying a difficult constraint. The rules are applied in this order.

The content of positive feedback is determined by the underlying learning theory SQL-Tutor is based on [9]. A positive feedback message is similar to the negative one: it points to a part of the solution which is relevant, and explains the domain principle involved. A negative feedback message in addition discusses how the student's solution violates the domain principle. Regarding presentation, we decided to provide positive and negative feedback simultaneously. If a student's solution was incorrect, but still matched some of the rules presented, positive feedback was presented first, followed by negative feedback. Figure 2 illustrates a situation in which the student received some positive feedback about join conditions, and then negative feedback, on the *Error Flag* level, pointing the student to a mistake in the FROM clause.

4 Evaluation Study

An evaluation study was conducted at the University of Canterbury with students enrolled in an introductory database course, from May 9, 2007 to June 6, 2007. The participants logged in to SQL-Tutor for the first time during scheduled labs, but could use it later at any time before the end of the study. The participants were randomly assigned to either the control group or the experimental group. The control group students received only negative feedback, while experimental group received both negative and positive feedback, as explained in the previous section. Out of 79 students enrolled in the course, 55 logged into SQL-Tutor at least once, with 51 completing the (online) pre-test. The maximum mark on the pre-test was 4. There was no significant difference between the mean scores on the pre-test for the two groups. All student actions were recorded in the logs. Some students used the system for a very short time, and we have eliminated logs of those students who used SQL-Tutor for less than 10 minutes, as we assumed that this was insufficient time to produce learning effects. After eliminating these students, we were left with 41 participants who attempted the pre-test. Table 1 shows some statistics from the study.

The experimental group students attempted and solved almost the same number of problems in significantly less time than the students in the control group, and also made fewer attempts at problems. The students from both groups also acquired the same amount of knowledge, as measured by the number of constraints learned. The number of learned constraints is an internal measure based on a very simple heuristic.

The student model in SQL-Tutor stores the history of usage of each individual constraint. To see whether a student knows a constraint at the beginning of the interaction, we take the first five attempts on it. If the student satisfied that constraint more than seventy percent (70%) of times when it was relevant, then we assume the student knows the constraint. Otherwise, we assume that the student is yet to learn that constraint. At the end of the interaction with SQL-Tutor, we observe the last five attempts taken and perform the same calculation. If the student did not know the constraint initially, but has now satisfied that constraint more than 70% of the time, then the constraint has been learned. The measure therefore only applies to newly acquired constraints, learned as a result of using the system.

Table 1. Summary of students' interaction with SQL-Tutor

	Control	Experimental	p
Participants	23	18	
Pre-test mean (sd)	1.7 (0.8)	2.1 (1.3)	
Constraints learned	10 (6.1)	9.3 (6.8)	
Time (min)	193.8 (198.7)	92.3 (44.7)	0.012
Problems Attempted	28 (25)	26 (15)	
Problems Solved	25 (24)	22 (15)	
Total Attempts	119 (99)	98 (66)	
Time per Solved Problem	9.8 (7.9)	5.8 (4.8)	0.024
Time per Attempted Problem	7.5 (4.5)	4.1 (2.0)	0.002
Lab-test mean (%)	57 (26.5)	59.3 (24.3)	

We analysed the difference in means for the two groups for each of the measures reported in Table 1 using the t-test. The only significant differences are for the time spent with SQL-Tutor, average time per problem solved and the average time per attempted problem. The mean time for the experimental group is only half the amount of time used by the control group. While students in the experimental group spent on average 101.5 minutes less than students in the control group, they were able to solve almost the same number of problems; an average of 22 solved problems in the experimental group compared to an average of 25 by students in the control group. We also compared students' performance on a lab test, which was an assessment item for the course, performed after the SQL-Tutor study. The experimental group's mean performance on the lab-test was slightly higher, although the difference is not significant.

We also performed an ANCOVA analysis with time as the dependent variable, pre-test score as the co-variate and mode (group membership), number of negative feedback messages seen, number of solved problems and total attempts as explanatory variables. All factors together accounted for 86% (R^2=0.862) of the variability in total time and an interaction was noticed between mode and total time spent in the system ($p=0.005$). Also significant was the total number of attempts ($p<0.0001$). It is worth noting that prior knowledge as measured by pre-test was not a significant factor ($p=0.392$).

To further consider the impact of all factors on learning, we performed two multiple regression analyses. Of particular interest was the students' prior knowledge as reflected in the pre-test score, the total time spent receiving tutoring and the number of feedback messages seen. The last factor consisted of the number of

negative feedback messages seen by control group, while for the experimental group, we used the two factors: the number of negative and the number of positive messages. The number of learned constraints was used as a measure of learning. The results are shown in Table 2. The number of participants was low for this kind of analysis; however, we plan to repeat the study in 2008 and have a larger data set. Therefore the following results should be taken with caution. For the experimental group, the model we obtained including all four predictors accounted for 77% of the variance, with time contributing most. The number of positive feedback messages is the only (marginally) significant predictor, and accounts for 6% of the variance. For the control group, the model with three predictors explains 71% of the variance, and the number of negative feedback messages is the only significant predictor.

These results suggest that the students' prior knowledge did not affect the average number of constraints learned and did not explain a significant portion of the variance in the learning for any of the groups. We see that the strongest predictor of learning in the case of the experimental group is the number of positive feedback messages seen, while in the case of the control group, the strongest predictor turns out to be the number of negative feedback messages seen. It is not surprising that negative feedback is the strongest predictor of learning, for this is a rather common finding in educational research; however it is interesting that positive feedback turns out to be the strongest predictor for the experimental group. It shows that student's learning is influenced considerably by positive feedback messages received and that the more positive feedback messages a student receives, the more they are likely to learn (• =0.625). This evidence supports directly the hypothesis of this research; that positive feedback works by reducing uncertainty in student knowledge, decreasing the number of future errors by strengthening weak knowledge and in some cases creating new knowledge. The high correlation between learnt constraints and time also supports this, r=0.74 for the experimental and r=0.68 for the control group.

Table 2. Multiple regression results for constraints learned

		R^2 change	•
Control (R^2=0.714)	Pre-test	0.015 (ns)	0.083 (ns)
	Time	0.458 (p<0.001)	-0.285 (ns)
	Negative feedback	0.241 (p=0.001)	1.080 (p=0.001)
Experimental (R^2=0.766)	Pre-test	0.000 (ns)	-0.036 (ns)
	Time	0.560 (p=0.001)	0.075 (ns)
	Negative feedback	0.143 (p=0.021)	0.203 (ns)
	Positive feedback	0.063 (p=0.083)	0.625 (p=0.083)

We also performed a multiple regression of the lab-test scores using the same factors as a basis for comparison (Table 3). The models (containing all predictors) obtained account for 19% and 76% of the variance for the control and experimental group respectively. Unlike our previous findings with learned constraints, we found that student prior knowledge marginally significantly predicts lab-test scores for both groups. In the case of the control group, prior knowledge accounts for 13% of the variance. In the experimental group prior knowledge accounts for 57% of the variance but interestingly positive feedback is also marginally significant, accounting for 12% of the variance. Given the very low correlation between learned constraints and lab-test scores, and the findings of the learned constraint multiple regression analysis,

these findings are not at all surprising. They reveal that students sitting the lab-test are influenced strongly by other sources of knowledge outside of that taught by SQL-Tutor. Despite this however, positive feedback still appeared to be a significant factor again supporting our hypothesis that positive feedback increases learning.

Table 3. Multiple regression results for lab-test score

		R^2 change	•
Control (R^2=0.185)	Pre-test	0.134 (p=0.094)	0.375 (p=0.096)
	Time	0.002 (ns)	-0.402 (ns)
	Negative feedback	0.050 (ns)	0.502 (ns)
Experimental (R^2=0.763)	Pre-test	0.569 (p<0.001)	0.546 (p=0.096)
	Time	0.052 (ns)	0.394 (ns)
	Negative feedback	0.024 (ns)	-1.042 (ns)
	Positive feedback	0.118 (p=0.031)	0.885 (p=0.096)

Figure 3 shows the learning curves for both groups. These curves show the error rate averaged over all constraints and all students, for each occasion when constraints were used. The actual data are well approximated by the power curves. There is no significant difference between the learning rates of the two groups, which is consistent with the previous finding about the number of constraints learned. Please note that the learning curves show how students learn constraints as a function of the number of attempts they used the constraints, not as a function of time elapsed between attempts. The control group students simply needed more time to learn the same amount of knowledge.

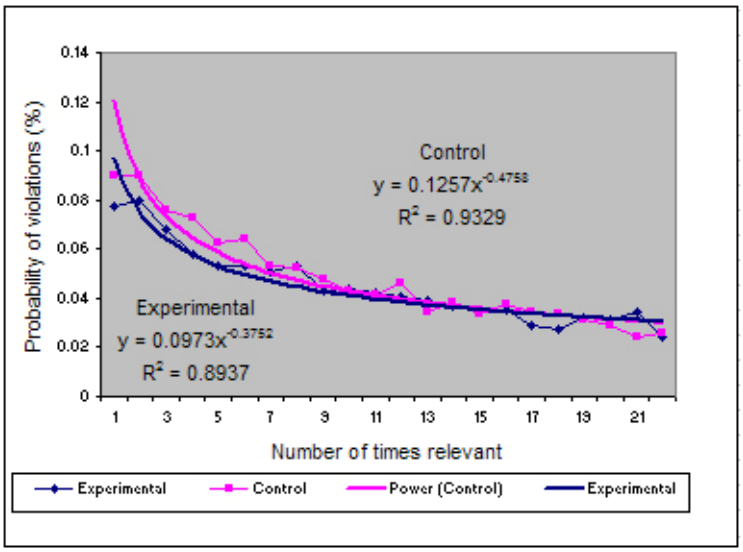

Fig. 3. Learning curves for all constraints

5 Conclusions

Starting from the observation that expert human tutors use positive feedback often, we modified SQL-Tutor, an ITS that teaches SQL querying, to give positive feedback. The modified version of the system provides positive feedback to students in several situations: when they are unsure of their actions, after they successfully use a hint provided by SQL-Tutor (either requested or provided by the system), after learning difficult domain principles or completing problems correctly. We implemented a set of rules that take into account the student's solution, the student model and the current state of interaction, and generate positive feedback.

An evaluation study performed in an introductory database course showed that positive feedback does affect learning significantly. Although we have not observed a significant difference in the amount of knowledge learned while interacting with SQL-Tutor, the students who received positive feedback solved the same number of problems and learnt the same amount of knowledge as the students in the control group but in half the time of the control group. The difference in time is significant, thus proving the importance of positive feedback in ITSs.

Acknowledgements

The project presented in this paper has been supported by a NZ Commonwealth MSc Scholarship to the first author. We thank all members of ICTG for their support and help.

References

1. Ohlsson, S.: Learning from Performance Errors. Psychological Review, **103**(2), 241-262 (1996)
2. Anderson, J.R.: Rules of the Mind. Lawrence Erlbaum (1993)
3. The Condition of Education, Technical Report 2007064, Institute of Educational Sciences, National Center for Education Statistics, Washington, DC, http://nces.ed.gov/pubs2007/2007064.pdf
4. Koedinger, K.R., Anderson, J.R.: Intelligent Tutoring Goes to School in the Big City. I.J. AIED, 8, 30-43 (1997)
5. Gertner, A., VanLehn, K.: Andes: A Coached Problem Solving Environments for Physics. In: Gauthier, G., Frasson, C. and VanLehn, K. (eds) ITS 2000. LCNS, Vol. 1839, pp. 131-142. Springer, Heidelberg (2000)
6. Mitrovic, A., Ohlsson, S.: Evaluation of a Constraint-Based Tutor for a Database Language. Int. J. Artificial Intelligence in Education, 10(3-4), 238-256 (1999)
7. Heffernan, N.T., Koedinger, K.: An Intelligent Tutoring System Incorporating a Model of an Experienced Human Tutor. In: Cerri, S.A., Gouarderes, G. and Paraguacu, F. (eds) ITS 2002. LCNS 2363, pp. 596-608. Springer, Heidelberg (2002)
8. Ohlsson, S., Di Eugenio, B., Chow, B., Fossati, D., Lu, X., Kershaw, T. C.: Beyond code-and-count analysis of tutoring dialogues. In R. Luckin, K. R. Koedinger and J. Greer (eds), Artificial Intelligence in Education: Building Technology Rich Learning Contexts That Work. pp. 349-356. IOS Press, Amsterdam (2007)
9. Zakharov, K., Mitrovic, A., Ohlsson, S.: Feedback Micro-engineering in EER-Tutor. In: C-K Looi, G. McCalla, B. Bredeweg, J. Breuker (eds) Proc. Artificial Intelligence in Education AIED 2005, IOS Press, pp. 718-725 (2005)