

Examples and Tutored Problems: How can Self-Explanation Make a Difference to Learning?

Amir Shareghi Najjar and Antonija Mitrovic

Intelligent Computer Tutoring Group
University of Canterbury, Christchurch, New Zealand
amir.shareghinajar@pg.canterbury.ac.nz
tanja.mitrovic@canterbury.ac.nz

Abstract: Learning from worked examples has been shown to be superior to unsupported problem solving in numerous studies. Examples reduce the cognitive load on the learner's working memory, thus helping the student to learn faster or deal with more complex questions. Only recently researchers started investigating the worked example effect in Intelligent Tutoring Systems (ITSs). We conducted a study to investigate the effect of using worked examples in combination with supported problem-solving in SQL-Tutor. We had three conditions: Examples Only (EO), Problems Only (PO), and Alternating Examples/Problems (AEP). After completing a problem, students received a self-explanation prompt that focused on the concepts used in the problem, to make sure that students acquire conceptual knowledge. On the other hand, examples were followed by self-explanation prompts that focused on procedural knowledge. The study showed that the AEP and PO conditions outperformed EO in learning gain, while AEP outperformed PO in conceptual knowledge acquisition. Therefore, interleaving examples with supported problems is an optimal choice compared to using examples or supported problems only in SQL-Tutor.

Keywords: worked examples, problem solving, self-explanation, intelligent tutors

1 Introduction and Related Work

Many studies have shown the worked example effect, in which students who study worked examples learn more than students involved in unsupported problem solving. Sweller et al. [1] explain the worked example effect based on the Cognitive Load Theory (CLT). They show that examples decrease the cognitive load on the learner's working memory. Thereby, learning from worked examples is more helpful for novices who have to deal with an enormous amount of cognitive load.

There has been no agreement on how much assistance should be provided to students during learning. Kirschner et al. [2] show that maximum assistance (e.g. examples) is more efficient than minimal assistance (e.g. unsupported problem-solving) which has been corroborated by prior studies like [3]. Recently researchers focused on

different example-based learning strategies. Van Gog et al. [4] investigate the difference between worked examples only (WE), worked examples/problem-solving pairs (WE-PS), problem-solving/worked examples pairs (PS-WE) and problem-solving only (PS) for novices. They found that the participants in the WE and WE-PS conditions had higher performances in the post-test than PS and PS-WE. Furthermore, the mental effort training and test rates in WE-PS and WE was lower than PS and PS-WE. In a later study, Van Gog [5] used Modelling Examples (ME) in two conditions PS-ME-PS-ME and ME-PS-ME-PS in the Frog Leap game. A modelling example is a type of example in which an expert illustrates the solution in a video format [6]. After these two sequences of training, students had to work on two tasks, of which the second one was not similar to training tasks. There was no difference in learning performance since the students learnt most after studying the second worked example.

Many prior studies addressed the advantages of example-based strategy against unsupported problem-solving. Koedinger and Alevan [7] criticised those because of the very different amounts of information provided to the two conditions (the unsupported problem-solving condition received no feedback upon submitting solutions). As the response to this criticism, Schwonke et al. [8] compared a standard cognitive tutor (Geometry Tutor) to a new version which was enriched with faded worked examples. Both conditions had the same amount of learning, but the faded example condition led to significantly reduced learning time.

Worked examples are beneficial in ITSs, especially for novices because they do not have adequate prior knowledge to solve problems, and examples can help them obtain the needed information. Therefore, it could be assumed that using a combination of examples and problem-solving might lead to a better result.

Using examples decreases the working memory load. If the freed working memory loads with germane load, learning will improve. One way to increase the germane load is to involve students in self-explanation (e.g. [9]). Self-Explanation (SE) is a metacognitive process in which students give explanations after studying learning materials [10]. Researchers have found evidence that students who generate explanations themselves learn more than students who receive explanations [11].

Few students self-explain spontaneously, and therefore SE prompts can be used to encourage students to explain examples to themselves. SE prompts can be of different nature, according to the knowledge they focus on. For instance, Hausmann et al. [12] compared justification-based prompts (e.g. "what principle is being applied in this step?") and meta-cognitive prompts (e.g. "what new information does each step provide for you?") with a new type called step-focused prompts (e.g. "what does this step mean to you?"). They found that students in the step-focused and justification conditions learnt more from studying examples than students in the meta-cognitive prompts condition. In another study, Chi and VanLehn [13] categorised SE as either procedural explanation (e.g. answer to "Why was this step done?"), or derivation SE (e.g. answer to "where did this step come from?").

McLaren and Isotani [14] compared examples only, alternating worked examples with tutored problem solving, and pure problem solving with the ITS. They conducted a study using the Stoichiometry Tutor and modelling examples. The examples were combined with SE prompts in order to involve students in thinking deeper about the

examples; the authors refer to such examples as interactive examples [14]. There was no difference in the post-test performance between the conditions, but the group that learnt from examples only had a significantly lower learning time. However, the examples were followed by SE prompts while the problems were not. The authors indicate that this result is interesting at least in some domains, under some conditions.

Our study continues the previous research on comparing learning from worked examples versus supported problem solving; similar to [14], we also investigate learning from Examples Only (EO), Problems Only (PO), and Alternating Examples/Problems (AEP). Since SE is a very effective strategy, we introduced SE prompts not only after examples (as in [14]), but also after problem solving. Our hypothesis is that students in the AEP condition will learn more than the other two groups, and students in the PO condition will learn more than the students in the EO condition ($AEP > PO > EO$). We also hypothesized that the EO participants would spend less time than the other two groups, as similar findings resulted from prior studies.

We describe our approach in Section 2. Section 3 presents the results of the study, while the conclusions and the directions of future work are presented in Section 4.

2 Study Design and Procedure

The studies discussed in the previous section were conducted in well-defined domains with well-defined tasks. We wanted to study learning from examples in a different context: defining queries in the Structured Query Language (SQL), which is a well-defined domain with ill-defined tasks [15]. Our study was conducted with SQL-Tutor, which is a constraint-based tutor [16] developed and maintained by the Intelligent Computer Tutoring Group (ICTG). SQL-Tutor complements classroom instruction; we assume that students learnt about SQL in lectures, and the system provides numerous practice opportunities. For this study, we developed three versions of SQL-Tutor in which students work with different combinations of examples and problems. In all the three conditions, students were presented with pairs of isomorphic examples and/or problems. That is, students who were in the EO and PO conditions worked with example-example and problem-problem pairs respectively. The students in AEP group interacted with example-problem pairs. There were 10 pairs in all conditions.

We designed 20 problems with ten different levels of complexity, based on the CD collection database, which is one of the databases available in SQL-Tutor. For a problem, SQL-Tutor provides the problem text only. A worked example consists of the problem text, the SQL statement that is the solution and an explanation.

In order to reinforce learning further, we provided SE prompts both after worked examples and after problems. We developed two types of SE prompts. Previous research [8, 17] showed that worked examples increase conceptual knowledge more than problem solving; therefore we provided Procedural-focused Self Explanation (P-SE) prompts after examples to make sure that students pay additional attention to procedural knowledge. P-SE prompts therefore complement learning from examples. On the other hand, working with the ITS is strongly focused on procedural knowledge [17] and therefore after solving problems, students were given Conceptual-focused

Self-Explanation (C-SE) prompts in order to ensure that students reflect on the concepts covered in the problem they just completed and acquire conceptual knowledge in that way. Both types of prompts require students to select an answer from a list of options. Figure 1 shows a screenshot of SQL-Tutor when the student has completed a problem, and was then given an SE prompt. In this situation the student's answer was incorrect, and the system provided a correction.

SQL-TUTOR		History	Log Ou
Problem 9	Find the names of artists and instruments they played in 'Someone to watch over me' or 'Summertime'.	What is the role of the IN predicate?	
SELECT	lname , fname, instrument	<input type="radio"/> A) It allows you to specify tables. <input checked="" type="radio"/> B) IN allows you to specify multiple values in the WHERE clause. <input type="radio"/> C) IN allows you to define attributes in the WHERE clause. <input type="radio"/> D) None of the above	
FROM	song, recording, performs, artist		
WHERE	performs.artist=artist.id and recording.id=performs.rec and song.id=recording.song and title IN ('Someone to watch over me','Summertime')	<p>No, we cannot define attributes in the WHERE clause. IN allows us to specify a condition in WHERE.</p>	
GROUP BY			
HAVING			
ORDER BY			

Fig. 1. A C-SE prompt after a problem is solved

Figure 2 shows a screenshot of a P-SE prompt which was provided after the student read an example. In this specific case, the student gave a correct answer which was confirmed by the system.

SQL-TUTOR		History	Log Ou
Example 10	Find the titles of songs and their composers (first name and last name) sung by artists whose last name is Gabriel or Davis. SELECT song.title, composer.fname, composer.lname FROM artist, song, song_by, composer, recording, performs WHERE recording.id=performs.rec and artist.id=performs.artist and artist.lname in ('Gabriel', 'Davis') and song.id=song_by.song and song_by.composer=composer.id;	Which option is equivalent to artist.lname in ('Gabriel','Davis')? <input checked="" type="radio"/> A) (artist.lname = 'Gabriel' OR artist.lname = 'Davis') <input type="radio"/> B) NOT (artist.lname = 'Gabriel' OR artist.lname = 'Davis') <input type="radio"/> C) (artist.lname = 'Gabriel' AND artist.lname = 'Davis') <input type="radio"/> D) NOT (artist.lname = 'Gabriel' AND artist.lname = 'Davis')	
Explanation	The IN predicate allows us to check whether the value of an attribute appears in the enumerated set of values.	Great!! That's exactly like using OR operator.	

Fig. 2. A screenshot of a P-SE after an example

The participants were 34 students enrolled in the Relational Database Systems course at the University of Canterbury. They learned about SQL in lectures beforehand, and needed to practice in the lab. The students did not receive any inducements for participating in the study, but we told them that working with our system may help them learn SQL. We informed them that they would see ten pairs of problems, and that the tasks in each pair are similar. When students know that the tasks in each pair are isomorphic, they may use them more efficiently.

The students were randomly allocated to one of the conditions, giving sample sizes of 12 in PO, 11 in AEP and 11 in EO. First, the students took a pre-test for 10 minutes. Once the students logged in, SQL-Tutor randomly allocated them to one of the conditions (EO, PO, or AEP). The students then had 90 minutes to work with the system. They could choose to take the post-test at any time during the learning phase to finish the experiment.

The pre-test had five questions, three of which were multiple-choice questions and two were problem-solving questions. The first and the second multiple-choice questions measured conceptual knowledge students had, while the third question measured procedural knowledge. For the fourth and the fifth questions, students had to write a query to answer the question. These two questions measured procedural knowledge and the problem-solving skill of the students. The post-test was similar to the pre-test with one extra question about the difficulty of the tasks. We asked students to answer this question: "How difficult was it for you to complete the tasks in this study?" Students rated the complexity of the tasks on the Likert scale from 1 to 5 (simple to difficult). The maximum score on both tests was 11.

3 Results

The basic statistics about the study are presented in Table 1. There was no significant difference between the pre-test performances of the three groups. ANOVA revealed a significant difference between the post-test results ($p = .02$). The Tukey post-hoc test showed that the performance of the EO group was significantly lower than the AEP group ($p = .02$) and marginally significantly lower than the PO group ($p = .09$), thus confirming our hypothesis. The students in all three conditions improved significantly between the pre- and the post-test, as shown by the paired t-test reported in the Improvement row of Table 1. Correlations between the pre- and post-test scores are also reported in Table 1, but only the PO condition had a significant correlation ($r = .69$).

There was also a significant difference between the mean learning times of the three groups ($p < .01$). The Tukey post-hoc test revealed that the EO group spent significantly shorter time than students in the AEP group and the PO group (both $p < .01$). The EO group participants were free to work with the system for the whole session, but spent much less time than the other two groups. This shows that the EO condition did not engage students like AEP and PO did. One potential explanation for this is that students overestimated their learning based on worked examples, and finished the tasks in a very short time.

There was a marginally significant difference between the three groups in the number of examples/problems they attempted ($p = .05$). The Tukey post-hoc test revealed that the EO group attempted more tasks than PO ($p = .1$) and the AEP group ($p = .07$).

The three groups also differed significantly in the normalised learning gain¹ ($p = .01$). The Tukey post-hoc test revealed that the EO group learnt significantly less than students in the AEP group ($p = .02$) and the PO group ($p = .03$). When we analysed normalised learning gains on the problem-solving questions in the pre/post-tests (questions 4 and 5), we found a significant difference between the groups ($p = .01$). As we expected, the students in the PO and AEP conditions performed significantly better than the students in the EO condition on problem-solving questions (Tukey post-hoc test: EO and PO $p = .01$, EO and AEP $p = .04$), because students in the EO condition were not given any problem-solving tasks during the learning phase.

Table 1. Basic statistics (* denotes the mean difference significant at the 0.05 level)

	PO (12)	AEP (11)	EO (11)	p
Pre-test (%)	41.67 (13.82)	48.76 (13.19)	44 (14.63)	.48
Post-test (%)	72.73 (13.98)	77.69 (16.57)	58.68 (16.57)	*.02
Improvement	* $p = .0$, $t = -9.8$	* $p = .0$, $t = -5.1$	* $p = .03$, $t = -2.4$	
Pre/post-test correlation	* $p = .01$, $r = .69$	$p = .49$, $r = .22$	$p = .43$, $r = .26$	
Learning time (min)	69.67 (11.16)	65.91 (14.53)	38.45 (16.14)	* $< .01$
Number of attempted problems	14.58 (5.11)	14.09 (5.10)	18.63 (3.23)	.05
Normalised learning gain	.54 (.19)	.55 (.31)	.21 (.35)	*.01
Problem solving gain	.64 (.27)	.58 (.42)	.19 (.37)	*.01
Conceptual knowledge gain	.29 (.39)	.77 (.41)	.54 (.47)	*.03
Procedural knowledge gain	.59 (.22)	.48 (.42)	.13 (.40)	*.01
Perceived task difficulty	3.50 (.80)	3.27 (.90)	2.82 (.75)	

We also analysed the students' conceptual and procedural knowledge separately. Questions 1 and 2 in the tests measured conceptual knowledge, while the remaining three questions focused on procedural knowledge. There was a significant difference on both conceptual and procedural normalised learning gain. The Tukey post-hoc test reveals that the AEP group learned significantly more conceptual knowledge than the PO group ($p = .02$). We think that examples helped the AEP students to acquire conceptual knowledge. The students in the AEP condition acquired the most conceptual knowledge since they saw both examples and C-SE prompts. That was the only significant difference revealed by the Tukey post-hoc test. There was also a significant difference in the procedural knowledge gain ($p = .01$); the Tukey post-hoc test revealed a significant difference was between the PO and EO conditions ($p = .01$), and a marginally significant difference ($p = .06$) between the AEP and EO conditions.

¹ Normalised learning gain = (Post test - Pre test) / (Max score - Pre test)

In the post-test we also asked students about the perceived task difficulty. The Man-Whitney U test indicated that the PO group ranked the problems as more difficult in comparison to the ranking by the EO group; the difference is marginally significant ($p=.053$). This result was expected as problems impose more cognitive load on the working memory than examples [1].

We calculated the effect size based on the normalised learning gain using Cohen's d , reported in Table 2. The effect sizes for both the AEP and PO conditions are large in comparison to the EO condition.

Table 2. The effect size on normalised learning gain between the groups

Conditions		Effect size
AEP	PO	.04
AEP	EO	1.01
PO	EO	1.15

The participants received C-SE prompts after problems and P-SE after examples. Therefore, the AEP group saw half of the C-SE prompts that PO students received, and also half of the P-SE prompts that the EO participants were given. We also analysed the SE success rates for the three conditions, which are reported in Table 3. We found no significant difference between AEP and PO in C-SE, and also no significant difference in P-SE success rate for the students in EO and AEP.

Table 3. SE prompts analysis (* denotes the mean difference significant at the 0.05 level)

	PO	AEP	EO	p
C-SE success rate (%)	88.50 (7.5)	92.84 (10.36)	N/A	.26
P-SE success rate (%)	N/A	77.69 (19.74)	71.36 (11.20)	.37

The students in the PO and AEP groups could select the feedback level² when they submitted their solutions, up to the complete solution (the highest level of feedback). Therefore, the participants could transform a problem-solving task to a worked example by asking for the complete solution. For that reason, we analysed help requests submitted for the problems given to the PO and AEP conditions.

Table 4. Maximum hint level analysis

	PO	AEP	
Second problem in pairs	1.08 (1.68)	1.54 (1.69)	$p = .51$
First problem in pairs	1.33 (1.56)		

Table 4 shows the mean number of problems for which the participants requested complete solutions. Looking at the second problem in each pair (the first row of Table

² SQL-Tutor offers six levels of feedback [16]

4), there was no significant difference in this respect between the PO and AEP conditions. Moreover, we did not see a significant difference in the number of times the PO students requested complete solutions for the first/second problem of each pair ($p = .39$). This result shows the participants from the PO/AEP groups have not converted their problems to worked examples.

Figure 3 depicts the relationship between the normalised learning gain and the learning time. Each data point on this graph represents the mean normalised learning gain of all students who completed their sessions by the specific time. For example, there were three participants from the EO condition who completed their session with SQL-Tutor 22 minutes into the study, and their normalised learning gains were 0.07, 0.11 and 0.14. The corresponding mean normalised learning gain at 22 minutes is therefore 0.11 (this corresponds to the third data point for the EO group). Although the fitted curve is an estimate only, this graph can be used for predicting normalised learning gains for longer learning sessions. The figure shows that the learning gains of the AEP and PO conditions are much higher than those of the EO group. In our study, the participants spent less than 90 minutes learning; the graph shows that the PO condition has the highest predicted learning gain over longer sessions.

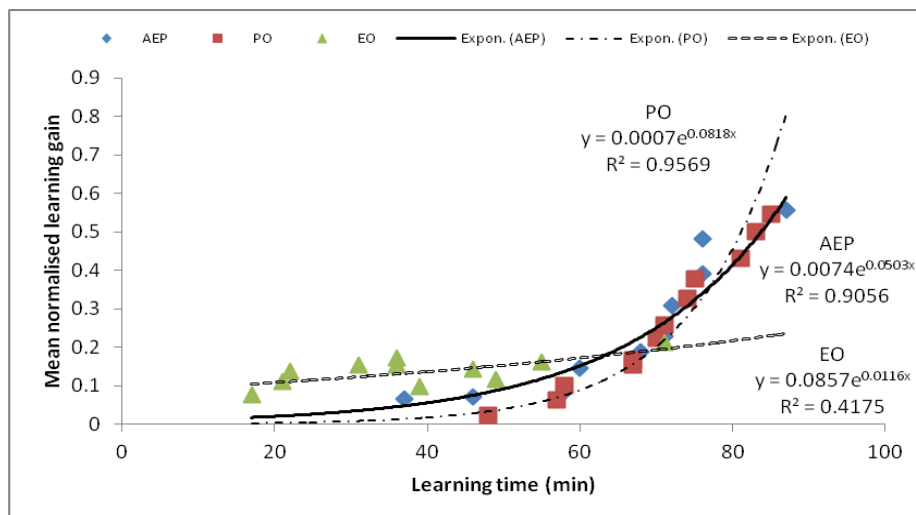


Fig. 3. Learning gain mean growth in time

4 Discussion and Conclusions

Our hypothesis was that the AEP condition would learn more than the PO and EO conditions, and PO would be superior to EO. Our analyses showed that the EO condition learnt significantly less than students in the other two conditions. All students had the same amount of time to work with the system, but the EO condition participants spent a significantly shorter time on reviewing examples. As stated previously, a possible explanation is that the participants could not accurately assess

their knowledge after reading examples, even with the addition of scaffolded self-explanation. As worked examples do not engage students like problems do, it is necessary to use some additional techniques to engage students to reason deeply about examples. This corroborates our previous finding that students who studied examples learnt less than students who solved problems.

Our results are in contrast with the findings presented in [14]. There are three main differences between the two studies. First, in our study the participants were given self-explanation prompts after problems, not only after worked examples (as in [14]). Moreover, we designed SE prompts to complement problem solving and examples. We provided procedural SE prompts after examples, as examples have been shown to reinforce conceptual knowledge more than procedural knowledge. We also provided conceptual SE prompts after problem solving to reinforce the acquisition of conceptual knowledge. Therefore, both types of SE prompts were designed so to complement the type of learning provided by the main activity (problem solving or learning from examples). The second difference is in the instructional domain used in each study. The instructional task in the McLaren and Isotani's study was simpler, consisting of simple algebraic equations and basic chemistry concepts, while in our study the participants were solving ill-defined design tasks. Thirdly, our constraint-based tutor provided feedback on demand while the Stoichiometry tutor used in [14] provided immediate feedback.

Why are worked examples not as effective as supported problem solving? Worked examples alone do not engage students as much as problem solving, and over time some students become less motivated to put enough effort into learning. Moreover, supported problem solving in contrast with unsupported problems avoid impasses, and is thus less frustrating and more effective. Examples may also induce an illusion of understanding after a certain number of tasks. For instance, students may think they have already learnt the example while they have not; consequently, they pass over the example very fast without spending enough time to process it which causes shallow learning. One potential approach to scaffold learning from worked examples is to provide support for self-assessment like in [18].

We found no significant difference between PO and AEP in the normalised learning gain and learning time. However, the AEP group acquired significantly more conceptual knowledge than the PO group. Consequently, the best instructional condition in our study was AEP, and our hypotheses were confirmed. The AEP participants learnt from the worked examples (the first task in each pair); when they were presented with isomorphic problems, they were already primed and did not have to deal with many unfamiliar details like students in the PO group.

Our study showed that learning from alternating examples with problems is superior to learning from problems or examples only, when the sequence of problems/examples is fixed. The results suggest that instead of just providing problem-solving opportunities, ITSs may provide worked examples followed by isomorphic problem solving. We recently conducted an eye-tracking study to investigate how students process examples. Based on the results of that study, our future research will focus on adding adaptivity to learning from worked examples in ITSs.

References

1. Sweller, J., Ayres, P., Kalyuga, S.: Cognitive load theory. Springer (2011).
2. Kirschner, P.A., Sweller, J., Clark, R.E.: Why Minimal Guidance During Instruction Does Not Work: An Analysis of the Failure of Constructivist, Discovery, Problem-Based, Experiential, and Inquiry-Based Teaching. *Educational Psychologist*. 41, 75–86 (2006).
3. Atkinson, R.K., Derry, S.J., Renkl, A., Wortham, D.: Learning from Examples: Instructional Principles from the Worked Examples Research. *Review of Educational Research*. 70, 181–214 (2000).
4. Van Gog, T., Kester, L., Paas, F.: Effects of worked examples, example-problem, and problem-example pairs on novices' learning. *Contemporary Educational Psychology*. 36, 212–218 (2011).
5. Van Gog, T.: Effects of identical example–problem and problem–example pairs on learning. *Computers & Education*. 57, 1775–1779 (2011).
6. Van Gog, T., Rummel, N.: Example-Based Learning: Integrating Cognitive and Social-Cognitive Research Perspectives. *Educational Psychology Review*. 22, 155–174 (2010).
7. Koedinger, K., Alevan, V.: Exploring the Assistance Dilemma in Experiments with Cognitive Tutors. *Educational Psychology Review*. 19, 239–264 (2007).
8. Schwonke, R., Renkl, A., Krieg, C., Wittwer, J., Alevan, V., Salden, R.: The worked-example effect: Not an artefact of lousy control conditions. *Computers in Human Behavior*. 25, 258–266 (2009).
9. Hilbert, T.S., Renkl, A.: Learning how to use a computer-based concept-mapping tool: Self-explaining examples helps. *Computers in Human Behavior*. 25, 267–274 (2009).
10. Chi, M.T.H., De Leeuw, N., Chiu, M.H., LaVancher, C.: Eliciting self-explanations improves understanding. *Cognitive science*. 18, 439–477 (1994).
11. Alevan, V., Koedinger, K.R.: An effective metacognitive strategy: Learning by doing and explaining with a computer-based Cognitive Tutor. *Cognitive Science*. 26, 147–179 (2002).
12. Hausmann, R., Nokes, T., VanLehn, K., Gershman, S.: The design of self-explanation prompts: The fit hypothesis. *Proceedings of the 31st Annual Conference of the Cognitive Science Society*. pp. 2626–2631 (2009).
13. Chi, M.T.H., VanLehn, K.A.: The content of physics self-explanations. *The Journal of the Learning Sciences*. 1, 69–105 (1991).
14. McLaren, B., Isotani, S.: When Is It Best to Learn with All Worked Examples? In: Biswas, G., Bull, S., Kay, J., and Mitrovic, A. (eds.) *Artificial Intelligence in Education*. pp. 222–229. Springer Berlin / Heidelberg (2011).
15. Mitrovic, A., Weerasinghe, A.: Revisiting Ill-Definedness and the Consequences for ITSs. In: Mizoguchi, V.D.R. (ed.) *The 14th International Conference on Artificial Intelligence in Education*. pp. 375–382. IOS Press, Amsterdam, The Netherlands (2009).
16. Mitrovic, A.: An Intelligent SQL Tutor on the Web. *International Journal of Artificial Intelligence in Education*. 13, 173–197 (2003).
17. Kim, R.S., Weitz, R., Heffernan, N.T., Krach, N.: Tutored Problem Solving vs . “ Pure ” Worked Examples. *Proceedings of the 31st Annual Conference of the Cognitive Science Society*. Austin, TX: Cognitive Science Society. 3121–3126 (2007).
18. Roll, I., Alevan, V., McLaren, B., Koedinger, K.: Metacognitive Practice Makes Perfect: Improving Students Self-Assessment Skills with an Intelligent Tutoring System. *Artificial Intelligence in Education*. pp. 288–295 (2011).