Wong, L.-H. et al. (Eds.) (2013). Proceedings of the 21st International Conference on Computers in Education. Indonesia: Asia-Pacific Society for Computers in Education

# Do novices and advanced students benefit differently from worked examples and ITS?

**Amir SHAREGHI NAJAR*, Antonija MITROVIC**
*ICTG, University of Canterbury, New Zealand*
*amir.shareghinajar@canterbury.ac.nz

**Abstract:** Prior research shows that novices learn more from examples than unsupported problem solving. Intelligent Tutoring Systems (ITS) support problem solving in many ways, adaptive feedback being one of them. However, when students repeatedly request hints from ITSs, problem solving is eventually replaced with worked examples when students request solutions to the current step or the whole problem. We conducted a study to observe the difference in learning outcomes when novices and advanced students learn from examples or with an ITS. The study had three conditions: Examples Only (EO), Problems Only (PO) and Alternating Examples and Problems (AEP). After each example/problem, students received Self-Explanation (SE) prompts. The result shows that novices learnt significantly more conceptual knowledge in the AEP compared to the PO condition. Moreover, novices in the AEP and PO conditions performed significantly better on SE prompts than students in the EO condition. Advanced students who learnt from examples only did not significantly improve in the study. Overall, the study suggests using AEP for novices and either AEP or PO for advanced students. The results clearly reveal that using examples alone is not an effective approach for novices and advanced students in comparison with ITSs.

**Keywords:** Worked examples, problem solving, intelligent tutoring systems, novice and advanced students

## 1. Introduction

Students with limited prior knowledge struggle with problem-solving. Human tutors often provide worked examples to novices as a way of providing missing knowledge. Numerous studies performed over the last three decades have proven the advantage of worked examples over unsupported problem solving (i.e. solving problems without guidance or feedback) (Sweller, 2006; Atkinson et al., 2000).

On the other hand, Intelligent Tutoring Systems (ITS) support problem solving by providing adaptive scaffolding in terms of feedback, guidance, problem selection and other types of help. Recently several studies have compared learning from examples to learning with ITSs (Schwonke et al., 2009; McLaren and Isotani, 2011; Kim et al., 2007). However, little attention has been devoted so far to the difference between novices and advanced students in those two types of situations.

We conducted a study that compared learning from examples only (EO), alternating examples and tutored problems (AEP), and tutored problems only (PO) in the area of specifying database queries in SQL (Shareghi Najar and Mitrovic, 2013). In this paper, we explain the results that show how advanced students and novices performed in that study. Our hypothesis is that novices and advanced students would not learn more from worked examples in comparison with ITSs or a mixture of examples and ITSs.

We start by presenting a short overview of related work, followed by a description of our approach in Section 3. Section 4 presents the results of the study, while the conclusions and the directions of future work are presented in Section 5.

## 2. Related Work

Many prior studies have shown the worked example effect: students learn more from worked examples than unsupported problem solving. Sweller et al. (2011) explain the worked example effect underlying the Cognitive Load Theory (CLT).

Sweller (2006) identifies three different loads for the working memory: intrinsic, extraneous and germane load. Intrinsic load is caused by the nature and difficulty of the learning task; as much as a problem is more complex, its intrinsic load is higher. Extraneous load is caused by information which is not related to learning like noise in the class or an unrelated joke during the lecture. In contrast to the extraneous load, germane load is caused by information which is related to learning materials.

Clark et al. (2006) outline different strategies and instructions to reduce extraneous load, and increase the germane load. Examples reduce the cognitive load on the working memory; thereby, learners acquire more knowledge from examples than unsupported problem solving. Examples provide information learners need to solve a problem. As a result, novices who normally do not have enough prior knowledge to solve problems benefit more from examples than unsupported problem solving (Van Gog and Rummel, 2010; Sweller and Cooper, 1985).

There has been no agreement on how much assistance should be provided to students. Kirschner, Sweller and Clark (2006) show that maximum assistance (e.g. examples) is more efficient than minimal assistance (unsupported problem solving) which has been corroborated by prior studies (Atkinson et al., 2000). Apart from the advantages of examples versus unsupported problem solving, recently researchers focused on different example-based learning strategies. van Gog et al. (2011) 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) on novices. The results show that the participants in WE and WE-PS had a higher performance in the post-test than PS and PS-WE. Furthermore, the mental effort imposed by WE-PS and WE was lower than PS and PS-WE.

The question is whether using examples for novices or advanced students is the best approach in comparison to Intelligent Tutoring Systems (ITS)? In contrast to unsupported problem solving, ITSs never leave students at impasse, because students can ask for different level of hints including final solutions. Research has shown that ITSs improve learning significantly more than unsupported problem solving (Mitrovic, 2012; Vanlehn, 2011; Koedinger et al. 1997)

Kim and colleagues (2007) discuss two experiments on pure worked examples and ITS. This research was done in Statistics and both procedural and conceptual knowledge acquisition were measured. In the first study, there was no significant difference between advanced students and novices. The second experiment shows that worked examples improved learning in both conceptual and procedural knowledge, and the ITS significantly improved students procedural knowledge. The paper concludes that worked examples outperform the ITS on conceptual knowledge, and use less learning time; on the other hand, the ITS is superior in procedural knowledge acquisition, but takes more time.

McLaren and Isotani (2011) compare examples only, alternating worked examples with tutored problem solving, and problem solving only. They conducted their study using Stoichiometry Tutor. Results show that students learnt the same from the three conditions, but students who worked with examples only used less time than the other two groups. However, examples were followed by Self-Explanation (SE) prompts while the problems were not. SE is a metacognitive process in which students give explanations after they study learning materials (Chi et al., 1994).

ITSs provide students with problems-solving tasks, and generally provide multiple levels of adaptive feedback, which differ in the amount of information provided. Many ITSs allow the student to select the level of feedback they want, and/or to ask for multiple feedback messages. Therefore, it is possible for the student to ask for feedback repeatedly, which eventually includes the solution for the current problem. In that way, the student can convert problem solving to learning from worked examples. Consequently, students working with ITSs get the benefit of learning from worked examples, and therefore could outperform those students who learn from examples only, because the ITSs provides students additionally with opportunities to try out the newly acquired knowledge in solving new problems.

## 3. Study Design

As far as we know, all prior studies that addressed the difference in learning from worked examples and ITSs used well-defined-tasks (e.g. geometry, algebra, and stoichiometry). Well-defined tasks are those for which there is an algorithm for solving problems (e.g. mathematics, physics) (Mitrovic and Weerasinghe, 2009). Therefore, it is interesting to observe how novices and advanced students would learn from examples in different context with ill-defined tasks. Rourke and Sweller (2009) show that the worked-examples effect can be obtained in both ill- and well-defined tasks compared to unsupported problem solving. To the best of our knowledge, learning from examples has never been compared with ITSs for ill-defined tasks. In our project, we focus on defining database queries using the Structured Query Language (SQL). Note that SQL is more complex than the learning tasks used in prior research.

We performed an experiment with SQL-Tutor, a constraint-based tutor that teaches SQL (Mitrovic, 2003). SQL-Tutor is a complement to traditional lectures; it assumes that the student has already acquired some knowledge via lectures and labs, and the tutor provides numerous problem-solving opportunities. The system currently contains more than 200 problems defined on 13 databases. Figure 1 illustrates the problem-solving page in SQL-Tutor, showing the problem text at the top, as well as the schema of the selected database at the bottom of the screen. Additional information about the meaning and types of attributes is available by clicking on the attribute/table name. The student can specify his/her solution by filling the necessary clauses of the SQL SELECT statement. Before submitting the solution to be checked, the student can select the level of feedback they want to receive. SQL-Tutor provides six levels of feedback. The lowest level, *Positive/Negative* feedback, simply states whether the solution is correct or how many mistakes there are. The *Error Flag* feedback identifies the clause that is incorrect. The *Hint* level specifies the message corresponding to one violated constraint, while the *Detailed Hint* provides more information about the relevant domain principle. The *Partial Solution* specifies the correct version of a clause that is incorrect in the student's solution, while the *Full Solution* provides the correct version of the SELECT statement. The feedback level automatically increases to the Detailed Hint level, while the student must explicitly request the higher levels. Students can attempt the same problem as many times as they want. Students may switch to another problem at any time. The system selects the next problem based on the student model.

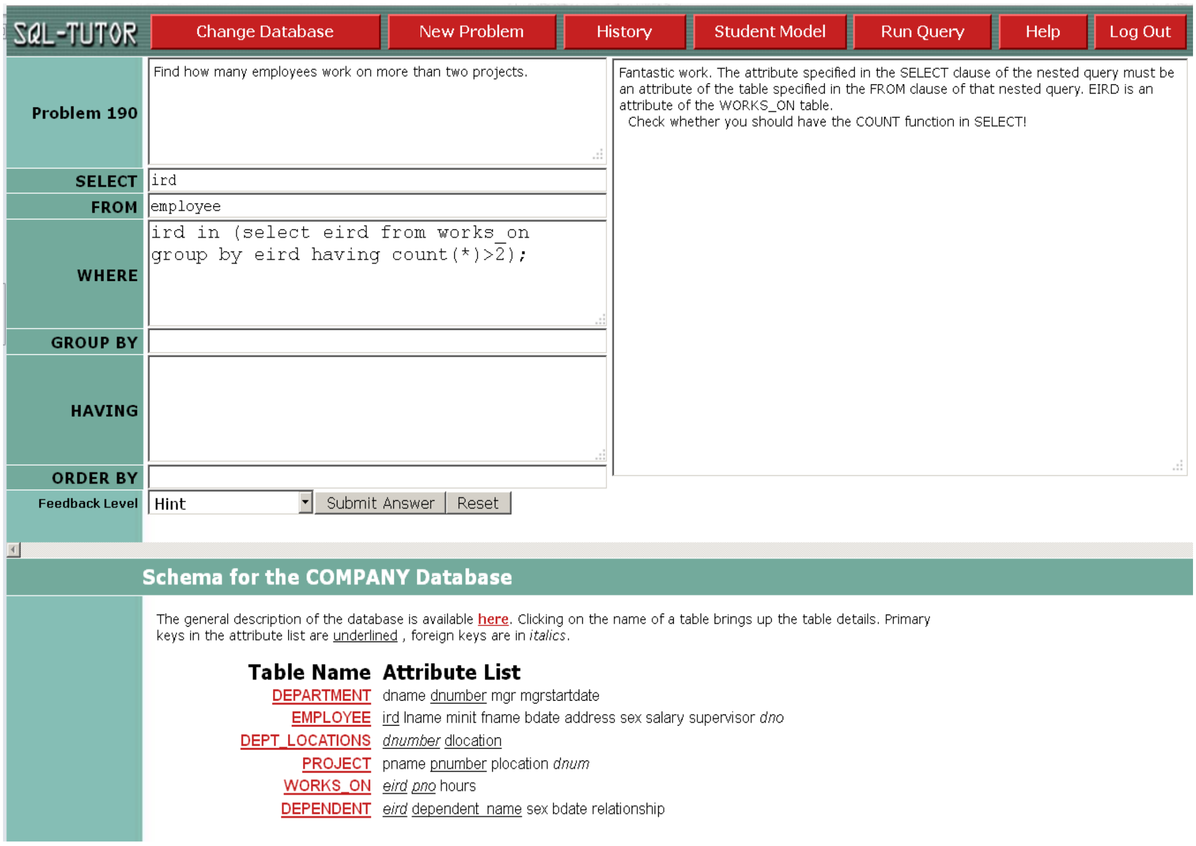


Figure 3. Screenshot of original SQL-Tutor

For this study, we developed three versions of SQL-Tutor which provided different combinations of worked examples and problems. Figure 2 shows the study design. In each condition, the student was given a set of 20 problems/examples arranged in pairs, so that the problems/examples in one pair were isomorphic. The Examples Only (EO) and Problems Only (PO) conditions presented isomorphic pairs of worked examples and problems consecutively, while the Alternating Examples Problems (AEP) condition presented a worked example followed by an isomorphic problem.

As discussed before, worked examples have been shown to decrease the load on the working memory. If the freed space in the working memory is used to increase germane load, learning should improve. Research has shown that an effective way to increase the germane load is to involve students in self-explanation (Hilbert and Renkl, 2009). Students who generate explanations themselves learn more than students who receive explanations (Aleven and Koedinger, 2002).

| **PO** | **AEP** | **EO** |
|---|---|---|
| n = 12 | n = 11 | n = 11 |
| Pre-test | | |
| 20 problems in 10 isomorphic pairs | 20 problems and examples in 10 isomorphic pairs | 20 examples in 10 isomorphic pairs |
| 1st in each pair: problem<br>2nd in each pair: problem | 1st in each pair: example<br>2nd in each pair: problem | 1st in each pair: example<br>2nd in each pair: example |
| Each problem followed by a C-SE prompt | Each problem followed by a C-SE prompt and each example followed by a P-SE prompt | Each problem followed by a P-SE prompt |
| Post-test | | |

Figure 4. Design of study with three conditions

Research on self-explanation shows that few students self-explain spontaneously (Chi reference), but can be encouraged to self-explain with carefully designed prompts. SE prompts can be of different nature, according to the knowledge they focus on. For instance, Hausmann et al. (2009) compare 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 (1991) categorized 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?"').

Previous research (Schwonke et al., 2009; Kim et al., 2007) showed that worked examples increase conceptual knowledge more than procedural knowledge, while problem-solving produces results in higher acquisition of procedural knowledge. To compensate for this, we developed two types of SE: Conceptual-focused Self Explanation (C-SE) and Procedural-focused Self-Explanation (P-SE). C-SE prompts encourage students to reflect on concepts of the learning material (e.g. "what does the select clause in general do?"). P-SE prompts encourage students to self-explain the procedures of solutions (e.g. "what will happen if we don't use DISTINCT in this solution?").

Figure 3 shows a screenshot of a situation when the student has finished reading an example. The complete example was shown at the same time. Next, the system shows a P-SE prompt, located on the right side of the screen. The student gives a correct answer to the prompt, and the system provides positive feedback.

Figure 4 shows a screenshot of a problem-solving task. In this situation, the student was given a C-SE prompt after s/he solved the problem. The student gave a wrong answer to the C-SE prompt, and because there is only one attempt per SE prompt, the system showed the negative feedback and revealed the correct answer. Once students received SE feedback, they could continue with the next task.

The participants were 34 students enrolled in the Relational Database Systems course at the University of Canterbury. They learned about SQL in lectures before-hand, 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 were similar.

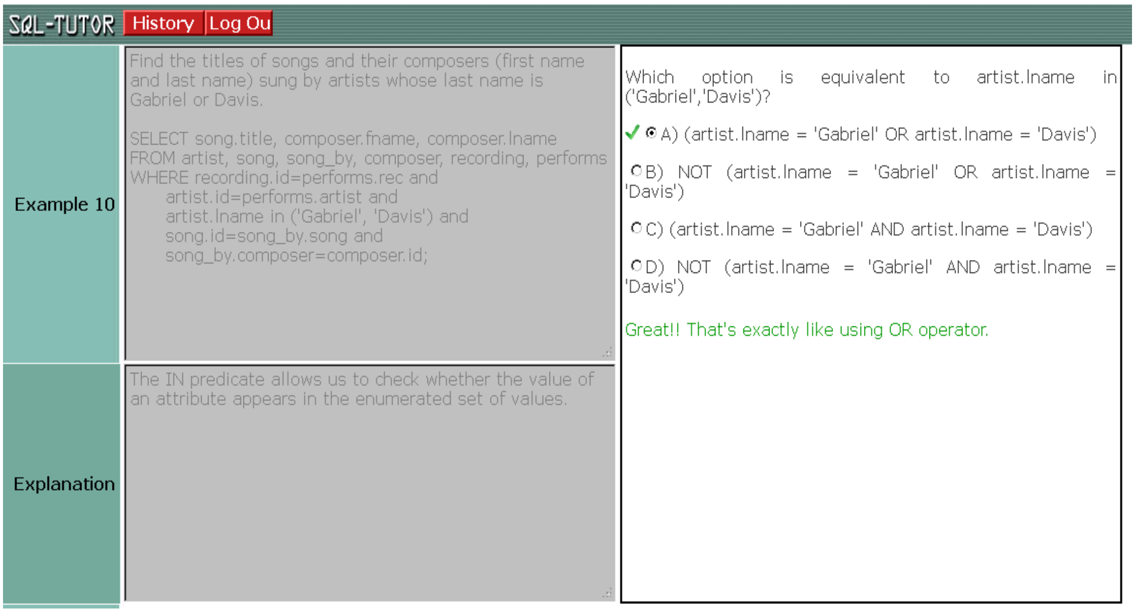


Figure 5. Screenshot of an example page followed by P-SE

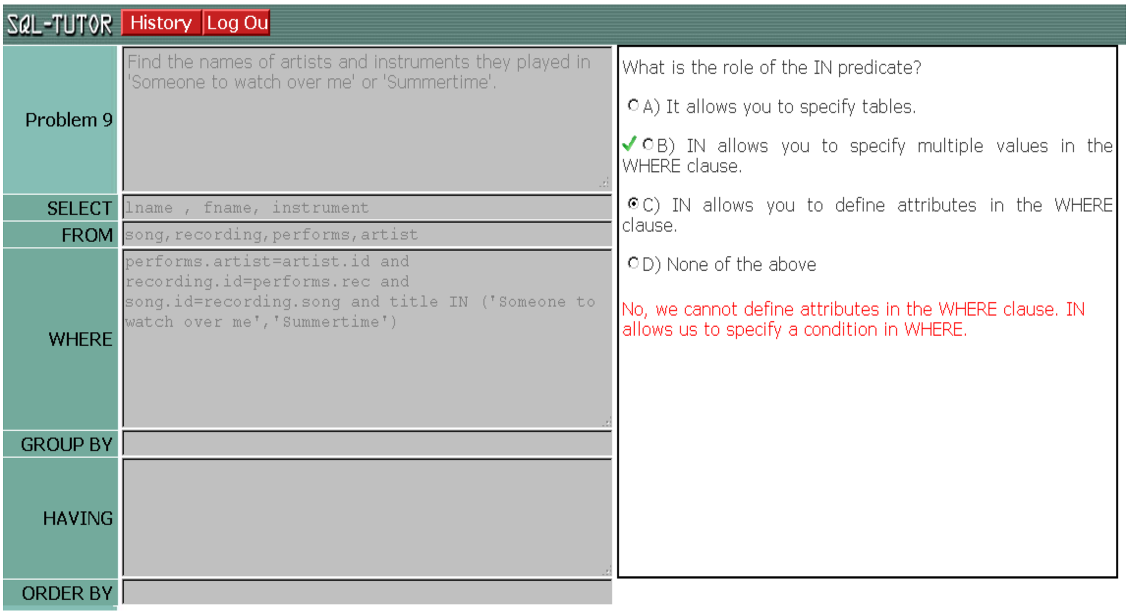


Figure 6. Screenshot of a problem solving page followed by C-SE

The study was conducted in a single, 90-minute long session. At the beginning of the session, 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), giving sample sizes of 12 in PO, 11 in AEP and 11 in EO. The students then interacted with SQL-Tutor, and took the post-test at the end of the session.

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 each test was 11.

## 4. Results

We calculated the average of scores in the pre-test and the post-test, and the time students spent on the system (Table 1). The students who had the pre-test scores lower than 45% were considered as novices and the rest were called advanced students.

Table 4. Basic statistics for all the participants

| | |
|---|---|
| Number of students | 34 |
| Pre-test (%) | 45 (14) |
| Post-test (%) | 70 (17) |
| Learning time (min) | 58 (20) |

Table 2 presents some statistics about the novice students. We used a significance level of .05 for all analyses. The Kruskal-Wallis 1-way ANOVA test did not reveal a significant difference on the pre-test performance of the three conditions; therefore, our groups were comparable. Using the same test, we saw no significant difference between the groups on the post-test. The Wilcoxon signed ranks tests show that novices in PO and AEP condition improved significantly between the pre-test and the post-test while EO shows a marginally significant improvement. There are no significant differences between the three conditions on the post-test performance and the normalized learning gain[1], but there is a marginally significant difference in learning time ($p = .06$). The Mann-Whitney test shows that novices in EO spent significantly less time than novices in AEP ($p = .03$) and PO ($p = .05$). The table also indicates a significant difference in the normalized conceptual knowledge gain ($p = .04$), and the Mann-Whitney test revealed that novices learnt significantly more conceptual knowledge from AEP than PO ($p = .01$).

Table 2. Dependent variables for novices ($^N$ Normalized)

| | PO | AEP | EO | p |
|---|---|---|---|---|
| Number of students | 6 | 5 | 5 | |
| Pre-test score (%) | 31 (11) | 36 (5) | 33 (10) | .79 |
| Post-test score (%) | 65 (12) | 73 (18) | 53 (14) | .14 |
| Improvement pre- to post-test | p = .03* | p = .04* | p = .07 | |
| Learning gain $^N$ | .50 (.14) | .56 (.30) | .29 (.20) | .13 |
| Learning time (min) | 67 (12) | 70 (12) | 46 (15) | .06 |
| Multiple choice questions $^N$ | .33 (.27) | .60 (.09) | .20 (.68) | .15 |
| Problem solving questions $^N$ | .56 (.22) | .55 (.44) | .24 (.35) | .28 |
| Conceptual knowledge $^N$ | .42 (.38) | 1.00 (0) | .70 (.45) | .04* |
| Procedural knowledge $^N$ | .52 (.20) | .45 (.36) | .18 (.29) | .12 |

[1] Normalized learning gain= (Post test - Pre test) / (Max score - Pre test), $^N$ represents normalized results in the tables.

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. The SE success rates for novices are reported in Table 3. The Kruskal-Wallis 1-way ANOVA test shows a significant difference between novices in the three conditions on overall success rate. The Mann-Whitney test reveals that novices in PO and AEP scored significantly higher than novices in EO ($p < .01$ and $p = .03$). Moreover, the Mann-Whitney test indicates a significant difference in P-SE success rate on SE prompts ($p = .3$); thus, novices in AEP performed significantly better than novices in EO who saw the same type of SE prompts (P-SE).

Overall, the analyses of the pre-test, post-test and SE performances confirm our hypothesis that novices benefit more from AEP or PO than using EO. We think that ITS engaged novices with both examples and problems while examples could not provide any rehearsal opportunity. On the other hand, AEP novices learnt significantly more conceptual knowledge than PO. Since novices in the PO condition did not have a chance to improve their conceptual knowledge (apart from C-SE prompts), the AEP novices outperformed PO by acquiring significantly more conceptual knowledge due to studying examples.

Table 3. Analysis of SE performance for novices

| | PO | AEP | EO | p |
|---|---|---|---|---|
| SE success rate (%) | 88 (7) | 87 (12) | 67 (8) | .02* |
| C-SE success rate (%) | 88 (7) | 90 (14) | N/A | .26 |
| P-SE success rate (%) | N/A | 85 (12) | 67 (8) | .03* |

Students who scored more than average in the pre-test were classified as advanced students, and their performance is reported in Table 4. The Kruskal-Wallis 1-way ANOVA reveals that there was no significant difference between the pre-test performances of the three groups; thus, our groups were comparable. Although the table shows no significant difference between the three conditions in the post-test, the Wilcoxon signed ranks tests revealed that advanced students in EO did not significantly improved between the pre-test and the post-test ($p = .42$). The table shows a marginally significant difference on the problem-solving, and a significant difference in learning time between the groups. The Mann-Whitney test shows a significant difference between EO and PO on problem-solving ($p = .04$), and learning time ($p <.01$). This result is in line with those studies show advanced students learn more from problem-solving only than reviewing examples only. The Mann-Whitney test also shows a significant difference between EO and AEP on learning time ($p = .02$). Note, that the result shows insignificant improvement between pre-test and post-test for students who studied examples only while students spent less time than the other groups on the system. That maybe cause by illusion of understanding.

Table 4. Dependent variables for advanced students ($^N$ Normalized)

| | PO | AEP | EO | p |
|---|---|---|---|---|
| Number of Students | 6 | 6 | 6 | |
| Pre-test (%) | 52 (6) | 59 (7) | 55 (10) | .16 |
| Post-test (%) | 80 (13) | 82 (15) | 64 (18) | .16 |
| Improvement pre- to post-test | p =.03* | p = .03* | p = .42 | |
| Learning gain | .59 (.24) | .55 (.36) | .15 (.46) | .23 |
| Learning time (min) | 73 (10) | 63 (17) | 32 (15) | <.01* |
| Multiple choice questions $^N$ | .17 (.26) | .50 (.44) | -.03 (.82) | .34 |
| Problem solving questions $^N$ | .72 (.32) | .61 (.45) | .16 (.42) | .08 |
| Conceptual knowledge $^N$ | .17 (.40) | .58 (.49) | .42 (.49) | .28 |
| Procedural knowledge $^N$ | .66 (.26) | .52 (.50) | .08 (.50) | .12 |

We analyzed the performance of advanced students on SE prompts, summarized in Table 5. Kruskal-Wallis 1-way ANOVA test shows no significant difference between the three groups. A possible explanation is that the difficulty of the self-explanation prompts was not suitable for the advanced students. The SE prompts gradually become more complex, but advanced students might not have difficulty understanding the prompts as they have more domain knowledge.

Table 5. Analysis of SE prompts for advanced students

| | PO | AEP | EO | p |
|---|---|---|---|---|
| SE success rate (%) | 89 (8) | 83 (12) | 75 (13) | .18 |
| C-SE success rate (%) | 89 (8) | 95 (6) | N/A | .22 |
| P-SE success rate (%) | N/A | 72 (24) | 75 (13) | .94 |

Overall, we found that novices improved the most from the AEP condition in comparison to the other two conditions. Moreover, advance students did not improve when learning from examples only; therefore, EO was not an appropriate approach for them. This is an interesting finding since the prior research suggests that students learnt the same from examples and ITSs.

The novices and advanced students in the PO and AEP groups could select the feedback level[2] 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. There was no significant difference, in number of requests for complete solution, between PO and AEP participants.

## 5. Conclusions

Prior research shows that students, particularly novices, learn more from examples than unsupported problem solving. On the other hand, most of the studies that compared examples with ITSs indicate that students learn the same from worked examples and ITSs, in domains with well-defined tasks. This encouraged us to observe the examples effect in a domain with ill-defined tasks (SQL) (Shareghi Najar and Mitrovic, 2013). We compared student performance in three conditions: alternating example/problems, problems only and examples only. In this paper, we discuss how novices and advanced students performed in that study.

The results show that novices who worked with alternating examples and problems and problems only outperformed novices who worked with examples only. This suggests that novices benefit most when they were engaged in tutored problem solving. On the other hand, the results show that novices in alternating examples and problems outperformed problems only in conceptual knowledge acquisition; thus, alternating examples and problems is the best learning strategy for novices. The difference between alternating examples and problems and the other two groups was that the novices were able to increase their initial learning by studying examples and then use what they have leant to tackle isomorphic problems.

In addition, advanced students did not significantly improve in the examples only condition. This is an expected result, since advanced students had enough prior knowledge, so what they need was practicing that knowledge in solving new problems. But the examples could not provide a problem-solving opportunity for the examples only group. Therefore, examples caused expertise reversal effect (Kalyuga, Chandler, and Sweller, 1998). Expertise reversal effect indicates that worked examples are more convenient in the early stages of learning while students could benefit more from problem solving in later stages (Salden et al., 2009).

Sweller and Cooper (1985) explained a two-step learning process. First, examples are suitable approach for students, particularly for novices, since examples reduce the cognitive load and increase the initial learning. Second, students use the cognitive schema created from reading examples in

[2] SQL-Tutor offers six levels of feedback (Mitrovic, 2003)

solving a similar problems. Our result is in line with two-step learning process. However, using ITS instead of examples leads to a higher performance, because ITS provides students with a variety of supports. In general, our study justified a learning strategy that helps students in early stages (novices) and in later stages (advanced students). This strategy suggests using a combination of examples and ITS for novices, then when student knowledge increases, the system can continue giving them a mix of examples and problem solving, or gradually switches to ITS only. Shareghi Najar and Mitrovic (2013) suggest that for a long learning time, problems only may even outperform alternating examples and problems condition since advanced students do not need any more knowledge, and what they need is applying those learned schemas in solving new problems.

Although the students learning from examples only could have spent more time to read and review examples, they preferred to finish the study early. They therefore took the post-test before learning enough; thus the examples caused illusion of understanding (Shareghi Najar and Mitrovic, 2013). Illusion of understanding occurs, because the students are reading examples too fast and not trying to explain to themselves; therefore, overestimate their knowledge about a concept. Using examples does not engage students as deeply as problem solving does.

As all of the prior research compared examples with ITSs in the domains with well-defined tasks, we investigated the examples effect in SQL which is a well-defined domain with ill-defined tasks. SQL-Tutor provides more complex problems than algebra, geometry and stoichiometry tutors. Nevertheless, the age groups of participants were different.

It could be argued that this result is due to differences between conceptual-focused self-explanation and procedural-focused self-explanation. As discussed previously, we use two different types of self-explanation prompts in order to reinforce examples and problems with the most suitable prompts. For instance, it is not appropriate to reinforce examples with conceptual-focused self-explanation prompts because examples have been shown to increase conceptual knowledge (Schwonke et al., 2009; Kim et al., 2007).

A limitation of our study is the small number of participants. It would therefore be interesting to see the results of a larger study.

In our future research on using examples in ITSs, we will draw on three perspectives: when to give examples, how to design examples, and how to scaffold examples. However, the features of the instructional domain need to be considered. We have recently conducted a study to see how students study examples using an eye tracker, and in our future work, we aim to use the findings from that study to implement pedagogical interventions to improve learning from examples.

## References


Aleven, V., & Koedinger, K. R. (2002). An effective metacognitive strategy: Learning by doing and explaining with a computer-based Cognitive Tutor. *Cognitive Science*, *26*(2), 147–179.

Atkinson, R. K., Derry, S. J., Renkl, A., & Wortham, D. (2000). Learning from Examples: Instructional Principles from the Worked Examples Research. *Review of Educational Research*, *70*(2), 181–214.

Chi, M T H, & VanLehn, K. A. (1991). The content of physics self-explanations. *The Journal of the Learning Sciences*, *1*(1), 69–105.

Chi, M.T.H., De Leeuw, N., Chiu, M. H., & LaVancher, C. (1994). Eliciting self-explanations improves understanding. *Cognitive science*, *18*(3), 439–477.

Clark, R. C., Nguyen, F., & Sweller, J. (2006). *Efficiency in learning: evidence-based guidelines to manage cognitive load*. Jossey-Bass.

Hausmann, R., Nokes, T., VanLehn, K., & Gershman, S. (2009). The design of self-explanation prompts: The fit hypothesis. *Proceedings of the 31st Annual Conference of the Cognitive Science Society* (pp. 2626–2631).

Hilbert, T. S., & Renkl, A. (2009). Learning how to use a computer-based concept-mapping tool: Self-explaining examples helps. *Computers in Human Behavior*, *25*(2), 267–274.

Kalyuga, S., Chandler, P., & Sweller, J. (1998). Levels of Expertise and Instructional Design. *Human Factors and Ergonomics Society*, *40*, 1–17.

Kim, R. S., Weitz, R., Heffernan, N. T., & Krach, N. (2007). 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.

Kirschner, P. A., Sweller, J., & Clark, R. E. (2006). 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*(2), 75–86.

Koedinger, K. R., Anderson, J. R., Hadley, W. H., Mark, M. A., & others. (1997). Intelligent tutoring goes to school in the big city. *International Journal of Artificial Intelligence in Education (IJAIED)*, *8*, 30–43.

McLaren, B., & Isotani, S. (2011). When Is It Best to Learn with All Worked Examples? In G. Biswas, S. Bull, J. Kay, & A. Mitrovic (Eds.), *Artificial Intelligence in Education* (Vol. 6738, pp. 222–229). Springer Berlin / Heidelberg.

Mitrovic, A. (2003). An Intelligent SQL Tutor on the Web. *International Journal of Artificial Intelligence in Education*, *13*(2-4), 173–197.

Mitrovic, A. (2012). Fifteen years of constraint-based tutors: what we have achieved and where we are going. *User Modeling and User-Adapted Interaction*, *22*(1-2), 39–72.

Mitrovic, A., & Weerasinghe, A. (2009). Revisiting Ill-Definedness and the Consequences for ITSs. In V. D. R. Mizoguchi (Ed.), *The 14th International Conference on Artificial Intelligence in Education* (pp. 375–382). Amsterdam, The Netherlands: IOS Press.

Rourke, A., & Sweller, J. (2009). The worked-example effect using ill-defined problems: Learning to recognise designers' styles. *Learning and Instruction*, *19*(2), 185–199.

Salden, R. J. C. M., Aleven, V., Schwonke, R., & Renkl, A. (2009). The expertise reversal effect and worked examples in tutored problem solving. *Instructional Science*, *38*(3), 289–307. doi:10.1007/s11251-009-9107-8

Schwonke, R., Renkl, A., Krieg, C., Wittwer, J., Aleven, V., & Salden, R. (2009). The worked-example effect: Not an artefact of lousy control conditions. *Computers in Human Behavior*, *25*, 258–266.

Shareghi Najar, A., & Mitrovic, A. (2013). Examples and Tutored Problems: How can Self-Explanation Make a Difference to Learning? *Proceedings of the 16h International Conference on Artificial Intelligence in Education*.

Sweller, J. (2006). The worked example effect and human cognition. *Learning and Instruction*, *16*(2), 165–169.

Sweller, J, Ayres, P., & Kalyuga, S. (2011). *Cognitive load theory*. Springer.

Sweller, John, & Cooper, G. A. (1985). The Use of Worked Examples as a Substitute for Problem Solving in Learning Algebra. *Cognition and Instruction*, *2*, 59–89.

Van Gog, T., Kester, L., & Paas, F. (2011). Effects of worked examples, example-problem, and problem-example pairs on novices' learning. *Contemporary Educational Psychology*, *36*(3), 212–218.

Van Gog, T., & Rummel, N. (2010). Example-Based Learning: Integrating Cognitive and Social-Cognitive Research Perspectives. *Educational Psychology Review*, *22*(2), 155–174.

Vanlehn, K. (2011). The Relative Effectiveness of Human Tutoring, Intelligent Tutoring Systems, and Other Tutoring Systems. *Educational Psychologist*, *46*(4), 197–221.