

EVALUATING THE BENEFITS OF WORKED EXAMPLES IN A CONSTRAINT-BASED TUTOR

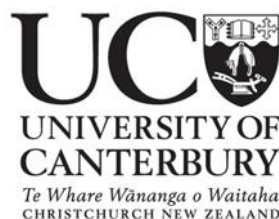
A thesis
submitted in partial fulfilment
of the requirements for the Degree
of
Doctor of Philosophy
by
Amir Shareghi Najar

Supervising Committee

Senior Supervisor: Professor Dr. Antonija Mitrovic
Associate Supervisor: Dr. Kourosh Neshatian
Associate Supervisor: Dr. Bruce M. McLaren

Examining Committee

Internal Examiner: Professor Dr. Judy Kay
External Examiner: Professor Dr. Gordon McCalla



University of Canterbury
April 2014

This thesis is dedicated to my mum, dad, and my dear wife.

Abstract

Empirical studies have shown that learning from worked examples is an effective learning strategy. A worked example provides step-by-step explanations of how a problem is solved. Many studies have compared learning from examples to unsupported problem solving, and suggested presenting worked examples to students in the initial stages of learning, followed by problem solving once students have acquired enough knowledge. Recently, researchers have started comparing learning from examples to supported problem solving in Intelligent Tutoring Systems (ITSs). ITSs provide multiple levels of assistance to students, adaptive feedback being one of them. The goal of this research is to investigate using examples in constraint-based tutors by adding examples into SQL-Tutor. SQL-Tutor is a constraint-based tutor that teaches the Structured Query Language (SQL). Students with different prior knowledge benefit differently from studying examples; thus, another goal of the research is to propose an adaptive model that considers the student's prior knowledge for providing worked examples.

Evaluation of this research produced promising results. First, a fixed sequence of alternating examples and problems was compared with problems only and examples only. The result shows that alternating examples and problems is superior to the other two conditions. Then, a study was conducted, in which a fixed sequence of alternating worked examples and tutored problem solving is compared with a strategy that adapts the assistance level to students' needs. The adaptive strategy determines the type of the task (a worked example, a faded example or a problem to be solved) based on how much assistance the student received in the previous problem. The results show that students in the adaptive condition learnt significantly more than their peers who were presented with the fixed sequence of worked examples and problem solving. The final study employed eye tracking and demonstrated that novices and advanced students study SQL examples differently. Such information can be used to provide proactive rather than reactive feedback messages to students' actions.

Table of Contents

Abstract	3
Table of Contents	4
List of Figures	7
List of Tables.....	9
Acknowledgments	11
Chapter 1. Introduction.....	12
1.1 Motivation.....	14
1.2 The problem and significance of the research	14
1.3 Thesis contributions.....	15
1.4 Thesis outline.....	17
Chapter 2. Intelligent Tutoring Systems	18
2.1 E-learning systems.....	18
2.2 Architecture of Intelligent Tutoring Systems	21
2.3 Methodologies for developing ITSs.....	24
2.4 SQL-Tutor.....	26
2.5 Self-Explanation (SE).....	33
Chapter 3. Using examples in learning.....	35
3.1 Cognitive Load Theory (CLT).....	35
3.2 Worked examples and modelling examples	37
3.3 Self-explanation effects in the example-based strategy.....	37
3.4 Learning from examples vs. unsupported problem solving	38
3.5 Learning from examples vs. tutored problem solving.....	42
3.6 Related work on adaptive worked examples	45
3.7 Is using examples always helpful?.....	47
3.8 Related work on using examples in the SQL domain	49
3.9 How should examples be delivered.....	50
3.10 Conclusion	54
Chapter 4. Eye tracking.....	56
4.1 Eye tracker, Tobii TX300.....	56
4.2 Eye-tracking literature review	60
4.2.1 Eye tracking for user classification (experts or novices).....	60
4.2.2 Eye tracking and ITSs.....	61
4.2.3 Eye tracking and learning from examples.....	63

4.3	Conclusion	64
Chapter 5.	First study: Worked-example effect in SQL-Tutor	65
5.1	Experiment design	66
5.2	Results	77
5.2.1	Results for all students	77
5.2.2	Learning curves	81
5.2.3	Results for novices and advanced students	82
5.3	Discussion and conclusion	85
Chapter 6.	Second study: Adaptive examples in SQL-Tutor	90
6.1	Study	91
6.1.1	Experiment design	91
6.1.2	Self-explanation	93
6.1.3	Calculating assistance score	97
6.1.4	Calculating cognitive efficiency	101
6.1.5	Adaptive strategy	102
6.2	Results	105
6.2.1	Analysing learning supported by the two conditions	106
6.2.2	Results for novices and advanced students	112
6.3	Discussion	115
6.4	Conclusion	118
Chapter 7.	Third study: Identifying learner differences in example processing.....	121
7.1	Experiment design	122
7.2	Participants and the procedure	123
7.3	Results	125
7.3.1	Grouping the students	125
7.3.2	Time spent studying examples	126
7.3.3	Sample quality	127
7.3.4	Eye Gaze Pattern Analysis (EGPA) coding scheme.....	128
7.3.5	Analysis of patterns	130
7.3.6	Analysis of behaviours	132
7.3.7	Analyses of eye-tracking data	134
7.3.8	Using machine learning classifiers	135
7.3.9	Heat maps.....	139
7.3.10	Gaze plots.....	143
7.4	Discussion	148
Chapter 8.	Conclusion	150

8.1	Overview of the project	150
8.2	Significant findings and contributions	152
8.3	Limitations	153
8.4	Future directions	154
References		156
Appendix A:	Examples and explanations	171
Appendix B:	The pre-tests and post-tests	179
Appendix C:	Information sheet	187
Appendix D:	Consent form	190
Appendix E:	Human ethics committee low risk approval	193
Appendix F:	List of publications.....	195

List of Figures

Figure 2.1 The relationship between the three types of e-learning systems.....	19
Figure 2.2 ITS components.....	21
Figure 2.3 Problem-solving environment in SQL-Tutor	27
Figure 2.4 Screenshot of feedback levels in SQL-Tutor.....	28
Figure 2.5 Screenshot of Open Student Model in SQL-Tutor.....	30
Figure 2.6 The architecture of web-based SQL-Tutor.....	31
Figure 2.7 Two constraints that checks the syntax.....	32
Figure 2.8 Constraint that checks the semantics	33
Figure 4.1 The Tobii TX300.....	57
Figure 4.2 A screenshot of the calibration quality screen.....	58
Figure 4.3 A sample screen with fixations captured over three seconds.....	59
Figure 4.4 An example of AOIs defined for the SQL-Tutor interface	60
Figure 5.1 The architecture of the system	67
Figure 5.2 The schema of the CD collection database.....	67
Figure 5.3 Design of the study with three conditions	68
Figure 5.4 Sample of a problem/example, and corresponding C-SE and P-SE prompts.....	69
Figure 5.5 Screenshot of a worked-example page	73
Figure 5.6: Screenshot of an example page followed by P-SE.....	74
Figure 5.7 Screenshot of a problem-solving page.....	75
Figure 5.8 Screenshot of a problem-solving page followed by C-SE	76
Figure 5.9 The probability of constraint violations for PO and AEP	82
Figure 6.1 Design of the study	92
Figure 6.2 Screenshot of an example page followed by P-SE.....	94
Figure 6.3 Screenshot of rating the effort after problem solving.....	95
Figure 6.4 Screenshot of a problem-solving page followed by C-SE	96
Figure 6.5 Plots of Equations 6.8, 6.9 and 6.10	100
Figure 6.6 Study flow	103
Figure 6.7 Diagram for cognitive efficiency and the model decisions.....	104
Figure 6.8 CE distribution from the previous study data.....	104
Figure 6.9 Learning curves for the experimental and control groups.....	109
Figure 6.10 CE changes diagram.....	111
Figure 7.1 Book database schema	122
Figure 7.2 Screenshot of the worked-example mode of SQL-Tutor	124

Figure 7.3 Pre-calibration screen.....	125
Figure 7.4 Average time (in seconds) spent on each example.....	127
Figure 7.5 Areas of interest on the screen	129
Figure 7.6 The W-Ladtree classifier.....	137
Figure 7.7 W-BFTree.....	137
Figure 7.8 Rule Induction	138
Figure 7.9 W-JRip.....	138
Figure 7.10 Heat maps for Example 1 (left) and Example 2 (right).....	140
Figure 7.11 Heat maps for Example 3 (left) and Example 4 (right).....	141
Figure 7.12 Heat maps for Example 5 (left) and Example 6 (right).....	142
Figure 7.13 Gaze pattern for an advanced student on Example 1	144
Figure 7.14 Gaze pattern for a novice student on Example 1	145
Figure 7.15 Gaze pattern for an advanced student on Example 6	146
Figure 7.16 Gaze pattern for a novice student on Example 6.....	147

List of Tables

Table 5.1 C-SE and P-SE compared to self-explanation used in in prior research.....	71
Table 5.2 Basic statistics for all students.....	77
Table 5.3 Basic statistics for the three conditions.....	78
Table 5.4 The effect size on normalised learning gain between the groups.....	80
Table 5.5 SE prompts analysis.....	80
Table 5.6 Maximum hint level analysis.....	81
Table 5.7 The number of constraints learnt.....	82
Table 5.8 Dependent variables for novices.....	83
Table 5.9 Analysis of SE performance for novices.....	84
Table 5.10 Dependent variables for advanced students.....	85
Table 5.11 Analysis of SE prompts for advanced students.....	85
Table 6.1 Assistance scores for different levels of help.....	97
Table 6.2 Maximum assistance score when students do not ask for high-level feedback.....	101
Table 6.3 Result of the equations' evaluation.....	101
Table 6.4 Decision table.....	103
Table 6.5 Basic statistics for all students.....	106
Table 6.6 Basic statistics for the two conditions.....	107
Table 6.7 Cognitive efficiency and mental effort analysis.....	108
Table 6.8 Analyses of SE prompts.....	109
Table 6.9 The effect size between the experimental group and the control group.....	110
Table 6.10 Summary of pairs.....	110
Table 6.11 Basic statistics for novices.....	113
Table 6.12 Cognitive load and cognitive efficiency analyses for novices.....	113
Table 6.13 SE prompts analyses for novices.....	113
Table 6.14 Basic statistics for advanced students.....	114
Table 6.15 Cognitive load and cognitive efficiency analyses for advanced students.....	115
Table 6.16 SE prompts analyses for advanced students.....	115
Table 7.1 Comparisons between the two clusters.....	126
Table 7.2 Quality of segments.....	128
Table 7.3 Average pattern frequencies.....	131
Table 7.4 Patterns recognised in each segment.....	133
Table 7.5 Average behaviour frequencies.....	134
Table 7.6 Eye-gaze metrics for novices and advanced students in D_{AOI}	135

Table 7.7 Eye-gaze metrics for novices and advanced students in W_{AOI}	135
Table 7.8 Eye-gaze metrics for novices and advanced students in E_{AOI}	135
Table 7.9 Features used.....	136
Table 7.10 Accuracy of classifiers.....	136

Acknowledgments

Thank you to my supervisor Professor Antonija Mitrovic, for all her invaluable guidance and making this period of study enjoyable. Thank you to my associate supervisors Dr. Bruce M. McLaren and Dr. Kourosh Neshatian, for all their helpful advice and valuable feedback on my thesis.

I wish to thank my loving and patient wife for enduring the inevitable financial constraints, for supporting and encouraging me during this project, and for taking an active interest in my research. I am also extremely grateful to my parents for their love and support throughout my life. Without their support, I would have never been able to achieve all these successes.

I am grateful for the valuable discussions with the past and present members of the Intelligent Computer Tutoring Group, especially Jay Holland, Amali Weerasinghe, Myse Ali Elmadani, Moffat Mathews and Atefeh Ahmadi. They made my PhD study a pleasant and memorable experience.

Thanks to my friends who managed to drag me away from the computer when things got too stressful! In particular, thank you Ehsan Tabatabaei Yazdi.

Thank you also to all the students who participated in the experiments during 2012 and 2013.

Chapter 1. Introduction

During the 20th century, the typical educational setting included a group of students and a teacher in a classroom. A strategy that teachers use is to give a lecture and ask students to do homework; in other words, students learn the necessary knowledge in a class then practise using this knowledge in solving problems. Because technology is improving, learning and education science, like the other sciences, has rapidly developed; consequently, new tools, methods and strategies have been proposed to facilitate the learning process. Tutoring systems are one type of learning tool that has helped students in their studies for the past three decades. Computer-Based Training (CBT) and Computer Aided Instruction (CAI) were the first computer-based tools for teaching. However, these systems were not individualised for students; that is, they behave in the same way for all students and the learner's ability was not considered (Beck & Stern, 1996).

An Intelligent Tutoring System (ITS) is an adaptive system that helps students by giving feedback according to the student's knowledge and learning ability. These systems normally consist of an interface to interact with students; a pedagogical module, which determines pedagogical actions' content and timing; a student model, which records the student's answers and performance; a domain knowledge model, which represents knowledge being taught by ITS; and an expert model, which is similar to the domain knowledge model in that it contains the material to be taught, but represents the knowledge in the same way as a domain expert would.

Model Tracing (MT) (Anderson et al., 1995) and Constraint-Based Modelling (CBM) (Mitrovic et al., 2001) are two leading approaches in developing ITSs. Both methods are used for developing tutors that have problem-solving environments.

Although many ITSs have been designed, only a few are in commercial use. SQL-Tutor is one of these ITSs, developed in the Intelligent Computer Tutoring Group (ICTG) at the University of Canterbury, to teach Structured Query Language (SQL) (Mitrovic, 1998; 2003). Problem solving is the main learning strategy in ITSs. Systems with a problem-solving strategy provide students with problems of different complexities and guide them to correct answers. This guidance can be via either positive or negative

feedback. Positive feedback is the tutor's response to correct student solutions and negative feedback is for incorrect solutions.

Researchers have studied the effects of worked examples in learning for almost three decades. For instance, Sweller (2006) disclosed why the example-based strategy is more efficient and van Gog and Rummel (2010) review different aspects of example-based learning. Most of these studies show the advantages of using examples in comparison with unsupported problem solving (i.e. where students do not receive any guidance while solving an unsupported problem solving).

Koedinger and Alevan (2007) argue that all prior studies compared unsupported problem solving with examples. In supported problem solving (as in ITSs), students were guided with feedback. Therefore, research was required to compare ITSs with examples. More recently, a few studies have shown that students who studied examples learnt faster than students who solved tutored problems (McLaren & Isotani, 2011; Salden et al., 2010), but the results are not conclusive. Schwonke et al. (2009) say that example-based learning is more efficient than an ITS, but not necessarily more effective. McLaren and Isotani (2011) show that learning from examples only is more efficient than tutored problems only or a mixture of tutored problems and examples in the stoichiometry domain.

The use of examples in education is not a new topic, but adding examples to ITSs has been suggested and needs to be researched more. Researchers are not sure whether or not using examples can induce a better learning gain than problem solving; moreover, a combination of examples and ITSs also can lead to a better result (Corbett et al., 2013). Therefore, the aim of this research is to investigate the worked-example effect in a constraint-based tutor enriched with examples, and propose an adaptive model to use examples in an ITS. The worked-example effect explains that students learn more from examples than solving unsupported problems.

This introductory chapter presents an overview of the thesis. Our motivation for this research is explained in Section 1.1. The significance of this research is discussed in Section 1.2. Section 1.3 introduces research questions, followed by our solution. Finally, a guide to the rest of the thesis is outlined in Section 1.4.

1.1 Motivation

Having a well-equipped education system is an essential need in any society or industry to foster its members or employees. An appropriate education system can help to improve the skills of the next generation, such as cultural behaviour, individual knowledge or social knowledge.

This research is motivated by a desire to propose a new model for using examples in ITSs in order to improve learning gains. Furthermore, we expand this area to constraint-based tutors by adding examples into SQL-Tutor. This opens a new window for further research in domains with ill-defined tasks. According to Mitrovic and Weerasinghe (2009), there is no algorithm to use to find a solution in ill-defined tasks. Moreover, the goal is also underspecified and it is difficult to evaluate solutions for correctness. For instance, there is no a deterministic way to evaluate essays. There might be several essays which are equally good. Therefore, worked examples for ill-defined tasks only reveal one solution.

In addition, prior studies have compared ITSs with either pure examples, or examples reinforced with self-explanation (SE) prompts. While SE is an effective factor in improving knowledge transfer, we believe that SE can be more effective if the prompts are chosen adaptively, corresponding to examples or problem solving. It is, however, necessary to examine whether or not using SE after examples and problem solving is beneficial.

1.2 The problem and significance of the research

The effect of worked examples is not a new topic. However, most of the prior studies compared examples with unsupported problem solving, and only recently have researchers started to compare examples with ITSs. While most studies show the improvement in learning time, there are a few studies indicating that using examples leads to a better learning gain. We have not found in the literature any evaluation of worked examples in constraint-based tutors, except Mathews and Mitrovic (2009). Furthermore, prior studies only compared the results of using worked examples with ITSs in domains with well-defined tasks.

Therefore, we were interested in researching the use of worked examples in a constraint-based tutor with ill-defined tasks. Furthermore, research shows that using examples is superior to unsupported problem solving, especially for novices. We wanted to find an adaptive approach that uses worked examples within an ITS and fosters students' knowledge.

1.3 Thesis contributions

Researchers have compared ITSs with examples in the past few years, yet results are not conclusive to determine whether a mixture of examples and problems is helpful or not. Because research has shown different results for using examples in ITSs, we explore the benefits of alternating examples with ITSs and propose an adaptive model for using individualised examples in SQL-Tutor. We experimentally tested the following four hypotheses:

- **Hypothesis 1:** Using alternating examples and problems, or problems only in SQL-Tutor is superior to studying examples only.
- **Hypothesis 2:** As students need different amounts of knowledge in different stages of learning, alternating examples and problems will be a better choice than using problems only, particularly for novices because they need more guidance in initial stages of learning.
- **Hypothesis 3:** As students need different amounts of knowledge in different stages of learning, an adaptive model for providing individualised examples will lead to a better learning time and learning gain in comparison with a fixed sequence of examples and ITS.
- **Hypothesis 4:** Novices and advanced students study SQL examples differently.

Now we explain the method and steps we took to investigate Hypotheses 1 and 2. Previous studies have shown that learning from worked examples is superior to solving unsupported problems. Examples reduce the cognitive load on the learner's working memory; thus, helping the student to learn faster or deal with complex problems. ITSs support problem solving in many ways, adaptive feedback being one of them. However, the student can repeatedly request hints from ITSs, eventually receiving the complete solution and thus replacing problems with worked examples. Only recently researchers

have started comparing worked examples with ITSs and several studies show studying worked examples results in faster learning. Chapter 5 explains the study we conducted to investigate the effects of studying examples only (EO) in comparison with problems only (PO) and alternating examples/problems (AEP). Our results show that, in contrast to prior studies, learning solely from examples is not as effective as solving problems or a mixture of examples and problems. In our study, novices learnt the most from AEP, but advanced students learnt the same from AEP and PO. Novices and advanced students learnt less from EO than AEP and PO. Therefore, interleaving examples with supported problem solving is an optimal choice compared to using examples or supported problems only in SQL-Tutor.

The method used to investigate Hypothesis 3 is as follows. Research shows that as novices have insufficient prior knowledge to solve problems without getting help, novices benefit more from studying worked examples than from solving problems (Sweller & Cooper, 1985). This suggests that students with different expertise levels need different levels of assistance. ITSs can provide different levels of assistance, from maximum assistance to minimum assistance. In Chapter 6 we present a study we conducted to compare a fixed sequence of worked examples and tutored problem solving with a model that adapts the assistance level to students' needs. The adaptive model decides on the level of assistance for the next task based on how much assistance students received in the current problem. Results show that students in the adaptive condition learnt significantly more than students who had the fixed sequence of worked examples and problems. Novices who worked with the adaptive model learnt faster than novices who worked with the fixed sequence of worked examples and problems, and advanced students who worked with the adaptive model learnt more than advanced students who worked with the fixed sequence of worked examples and problems.

Finally, we explain an overview of the steps we took to investigate Hypothesis 4. We used an eye tracker, which provides information about the user's eye gaze movements on a screen. For many years, eye tracking has been used in Human Computer Interaction (HCI) research. Similarly, research on computerised educational systems also relies heavily on students' interactions with systems, and therefore eye tracking has been used to improve learning. In order to get deeper insights about how students use examples,

in Chapter 7 we report on a study that used eye tracking to compare behaviour of novices and advanced students. The study was performed in the context of SQL-Tutor. We propose a new technique to analyse eye-gaze patterns named eye gaze pattern analysis (EGPA). We also analysed the number and duration of eye-gaze fixations on different areas of interest. In order to comprehend an SQL example, students require information about tables and their attributes. The information is available in the database schema; thus, if students pay attention to the database schema, they understand SQL examples better. We analysed students' eye movements data from different perspectives and the results show that advanced students paid more attention to the database schema than novices. This result will help researchers to build ITSs that can provide proactive rather than reactive feedback messages to students' actions.

1.4 Thesis outline

In Chapter 2, we briefly describe the fundamentals of ITSs and present a short overview on SQL-Tutor. We discuss SE as an effective scaffolding technique to improve learning. Chapter 3 reviews prior research that has used examples in learning. We then discuss the approaches for providing adaptive examples. In Chapter 4 we introduce eye tracking and the studies which employed eye tracking to improve learning from ITSs and learning from examples. Chapter 5 explains the study we conducted to compare learning from examples only with alternating examples and problems and problems only. Chapter 6 presents the evaluation study performed using an adaptive model that provides individualised examples or tutored problems for students. In Chapter 7, we explain the study we conducted using an eye tracker to observe how novices and advanced students learn examples. Finally, the conclusions and future work are given in the last Chapter 8.

In the course of this research, we have prepared and presented eleven publications, which are listed in Appendix F.

Chapter 2. Intelligent Tutoring Systems

Expert human tutors can help students produce a large learning gain (Bloom, 1984). A lack of resources, however, means that not all students in classrooms get access to expert human tutors. This encouraged researchers to look for alternative tools to aid students. With growing technology, educational tools and learning strategies have rapidly changed. These dynamic improvements helped researchers to develop systems which are as effective as human tutors (VanLehn, 2011).

Intelligent Tutoring Systems (ITS) are one class of learning tools. ITSs provide students with individualised feedback to their answers. Problem solving is a widely used strategy in building ITSs, while an expert human tutor uses different strategies depending on the situation. For instance, human tutors use a mixture of examples and problems in their teaching strategy and balance this combination to improve the student learning gain. In this thesis, we deal with two types of learning: learning from solving problems with ITSs, and learning from studying examples. In this chapter, we look at e-learning systems, then discuss ITS design and SQL-Tutor. Finally, we review prior work on self-explanation.

2.1 E-learning systems

Researchers have employed computers to provide enhanced educational support to students since the late 1970s. Most of the earlier work on computer-based systems was focused on pedagogies and behaviour to enable the systems to act like expert human tutors (Carbonell, 1970). Computer-Based Training (CBT) and Computer Aided Instruction (CAI) were the earliest systems to aid students in learning, but those systems were inflexible. These systems were unable to give individualised feedback based on the student's response and all students were presented with the same material. Later, some systems allowed students to see any content they wished by using a navigation control, while others forced students to follow the order of the material. The navigation control allows the student to skip the material and work on the concepts the student prefer to learn. Therefore, research was divided into three overlapping types of e-learning systems:

CBT, Adaptive hypermedia and ITS (Mathews, 2012). The relationship between the three types is shown in Figure 2.1.

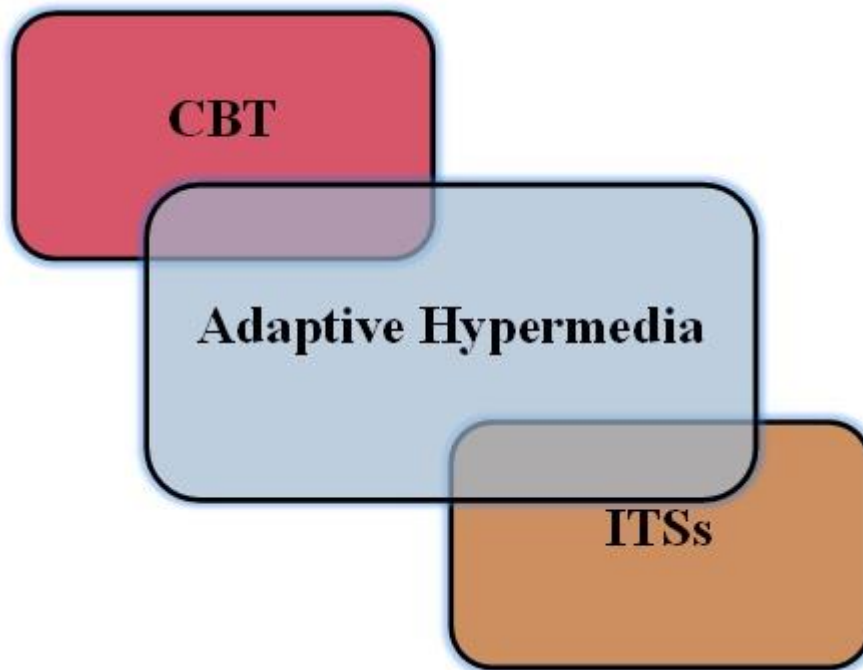


Figure 2.1 The relationship between the three types of e-learning systems

Virtual learning environments of today are CBT systems and the most popular of these are Learning Management Systems (LMSs). LMSs allow branching, in which students can access the domain content in any order. LMSs also allow for forums, quizzes and links to web pages and attachments. While students work with the system, an LMS collects the information in the background. Therefore, teachers can observe the performance of students. In the LMS, teachers organise the content and the system does not have any understanding of the content or the structure of the domain. A few CBTs are adaptive, such as Moodle (Modular Object-Oriented Dynamic Learning Environment)¹. Moodle has a *lesson module*² which has two types of pages: content pages and question pages. Question pages are the branches of the content pages, but the student's answers to a question can take the student to an entirely different page. Therefore, students may experience different paths through the knowledge base, depending on their answers (Brandl, 2005).

¹ <http://moodle.org>

² http://docs.moodle.org/22/en/Lesson_module

Adaptive hypermedia systems adapt in some way to course content, the student or both. An adaptive system might control the navigation by disabling a link when the student has not learnt the required materials (Brusilovsky, 2001). Navigation based adaptation can be implemented in different ways: *direct guidance* (e.g. a ‘next’ or ‘continue’ button); *adaptive sorting*, in which the system sorts the links of learning material according to the user model; *adaptive hiding*, which hides certain irrelevant or distracting links; and *link annotation*, in which the system provides more information by adding colour or icons to a link (Brusilovsky, 2001).

An ITS is an instructional system that recognises the learner’s behavioural patterns and provides a suitable response to those patterns (Beck & Stern, 1996). Therefore, understanding learners’ differences is often crucial. Different ITS components make teaching decisions and maintain a record for each student. Research has shown that ITSs lead to significant learning (Graesser et al., 2003; Suraweera & Mitrovic, 2004; Suraweera & Mitrovic, 2002) and even improve understanding similar to those resulting from expert human tutors (VanLehn, 2011).

Researchers have developed ITSs for numerous domains using a variety of approaches (e.g. algebra (Koedinger et al., 1997), geometry (Koedinger & Anderson, 1993), object-oriented software design using UML class diagrams (Baghaei et al., 2007) and capital investment (Mitrovic et al., 2008). Some ITSs have been developed for individual learning, while some are for collaborative learning. Model-tracing tutors (Anderson, 1996) provide problem-solving environments. In such environments, systems guide students towards the correct solution using immediate feedback. Simulation-based tutors teach the learning material through simulation environments (Alexe & Gecsei, 1996; Munro et al., 1997). Collaborative tutors have been developed to facilitate positive interaction among students by encouraging participation and supporting collaborative problem solving and tutoring between peers (Dillenbourg & Self, 1992). Similar to model-tracing tutors, collaborative tutors provide problem-solving environments in which students receive adapted feedback as they make progress towards correct solutions. Students who work with collaborative tutors are allowed to work at their own pace and request feedback when they need it.

2.2 Architecture of Intelligent Tutoring Systems

Similar to what is presented in (Beck & Stern, 1996), Figure 2.2 shows the main components in a typical ITS. The Domain Module contains facts and rules about the domain (domain knowledge), but they are represented in an understandable way for ITSs. An ITS uses the domain knowledge model to reason about the student's knowledge. The domain model can hold domain knowledge in the forms of procedural rules, constraints or frames (pages) of knowledge content. For instance, in Constraint-Based Modelling (CBM), the domain model contains all constraints that explain the domain.

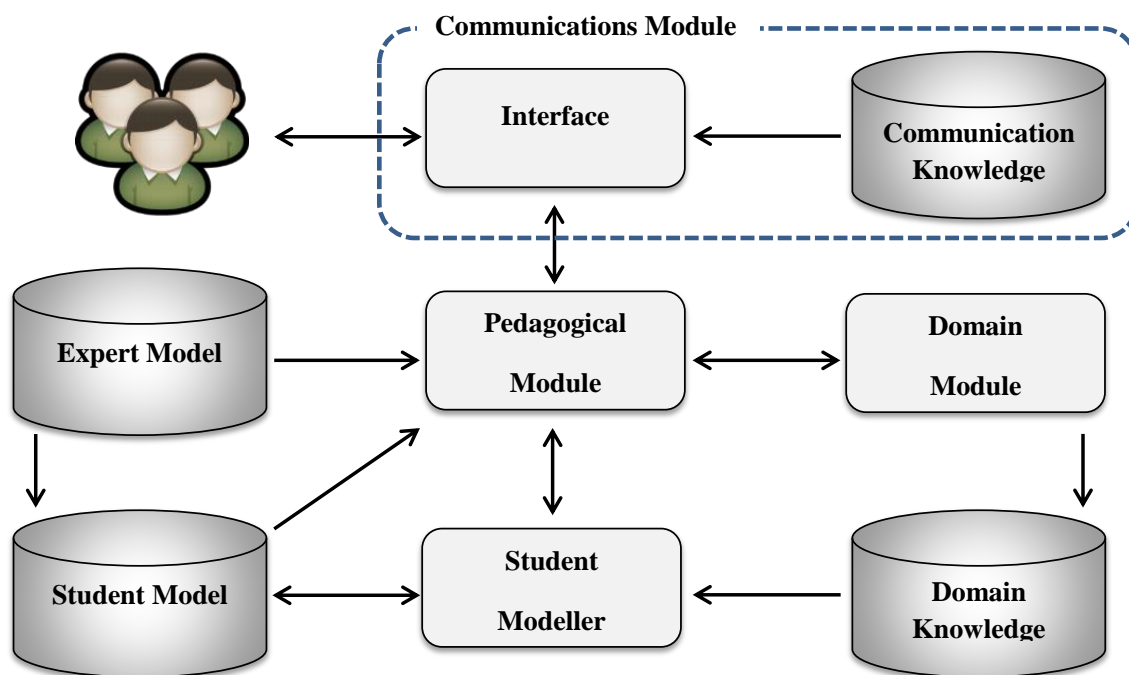


Figure 2.2 ITS components

The first domain modules were described as black box modules because although they could provide solutions to problems, they could not explain the reasons behind them. More recent ITSs have domain modules that are not only able to provide solutions, but also provide in-depth explanations to the solution process. For instance, model tracing tutors contain all the correct and incorrect solutions' steps and very specific feedback is attached to each step. That is, when the student makes an error on one step, the system can provide various levels of feedback. In CBM, each domain principle is described by one constraint. Thus, when a student makes an error, it means that one or more constraints are violated. Each constraint has multi-level feedback. Thus, when an error is made, the

student can get a variety of feedback corresponding to their error. We discuss in detail the model tracing and the constraint-based modelling in Section 2.3.

Both the expert model and the domain knowledge model contain knowledge about the domain, but with one significant difference: the expert model represents the knowledge in a way that an expert would. In some ITSs, the expert model is capable of solving problems (Clancey, 1979; Mitrovic, 2002); thus, sometimes the expert model is called the *problem solver*. The system can give feedback based on the difference between the student's solution and the expert model.

A student modeller is responsible for evaluating a student's solution and maintaining the student model. The student model shows the level of knowledge and skills the student has within that domain. Such information is used to make accurate pedagogical decisions. Student models are divided into two types: long and short term models (Mitrovic, 2003; Mitrovic & Martin, 2007). A long-term model represents general student characteristics such as name, level of expertise, history and the model of the student's knowledge (Mitrovic, 2003). Therefore, it shows an estimation of the expertise level of the student about domain concepts. Such information can be used to make larger pedagogical decisions. For instance, the system can choose the best next problem to solve. A short-term model represents the most recent interactions of the student with the system, which reflect upon the student's performance on the current task (Mitrovic, 2003). Therefore, the system uses such information to provide individualised feedback on what the student submitted to the system.

In most cases, student modelling representations are similar to domain modelling representations because student modelling is trying to determine the student's knowledge in terms of the domain. Model tracing and constraint-based modelling are two important representations that have been used with success.

There are many techniques for student modelling; we refer the ardent reader to (Greer & McCalla, 1994). In *overlay models*, the knowledge of the student is represented as a subset of the domain expert's knowledge; therefore, the initial state of an overlay model assumes that the student has no knowledge about the domain. As the student interacts with the system, the model becomes populated (Holt et al., 1994). The student model

could be empty when a student starts working with the system; thus, the system may not perform optimally. *Stereotype modelling* creates the student's domain knowledge with respect to the desired knowledge; thus, the model overcomes the problem of starting from an empty model. With stereotype modelling, students can be categorised based on their prior knowledge. For instance, a pre-test is one of the popular approaches, where the test can be as simple as asking the student to rate himself or herself as a novice or an advanced student (Rich, 1989). Another modelling technique is *knowledge tracing*, in which the system maintains an estimation of the probability that a rule has been learnt by the student. Based on the probability, the system can decide which rules have been mastered and which have not been practised (Corbett & Anderson, 1994). This paragraph is not a thorough list of student modelling techniques because variations and new methods are continually being developed.

The communications module controls all tutor-student interactions. Therefore, this module is responsible for the screen layout and the material representation. For example, it controls the graphical user interface and contains information about the type of communications with the student. It is important that an interface is intuitive and user-friendly as a complex interface may induce an unnecessary cognitive load into the student's working memory (Mayer, 2002). While the student works with the system, all the interactions are used to update the student model.

In an ITS, the pedagogical module makes all the teaching decisions by using the information from other components such as the domain module and the student model. For instance, considering the student model, the pedagogical module can decide which problem is appropriate for the student to solve. The pedagogical module contains pedagogical strategies, which define decisions that affect the student's learning. Most of the strategies are hard-coded into the system by developers. Because of the difficulty of adding new strategies, most ITSs have only one set of strategy for making each decision. Educators have used different pedagogical strategies, such as preventing harmful gaming of the system and using worked examples. When a student misuses the system's properties instead of using their knowledge to solve a problem, gaming the system happens (Baker et al., 2005). One example of gaming the system is when a student repeatedly asks for different levels of feedback until the system gives the complete

solution. Eventually the student changes the problem into an example (problem with solution). With preventing harmful gaming of the system strategy, the system avoids students misusing the system. For instance, an animated agent on the screen displays increasing levels of displeasure when the student games the system (Baker et al., 2006). In the worked-example strategy, students are given worked examples during the learning session. A worked example provides a step-by-step explanation of how a problem is solved (Gerjets et al., 2004). We discuss prior work about worked examples and different strategies for using examples in ITSs in Chapter 3.

2.3 Methodologies for developing ITSs

There are several approaches for developing ITSs such as Model Tracing (MT) (Anderson et al., 1995), Constraint-Based Modelling (CBM) (Mitrovic et al., 2001; Mitrovic et al., 2007) and Example tracing (Aleven et al., 2009). These three methods are used for developing tutors with problem-solving environments.

MT is based on the Adaptive Character of Thought-Rational (ACT-R) theory of cognition (Anderson, 1996). In the theory, a difference is made between procedural and declarative knowledge. For example, all the algebra laws that a student has studied about in the text book are declarative knowledge. Using declarative knowledge, the student knows how to apply the knowledge for solving a problem. Production rules specify which procedural knowledge is applied to solve a problem. Thus, a production rule is the relationship between a situation, a goal and an action. The action takes the person from the current situation to the goal.

Situation, Goal \rightarrow Action

Therefore, the production rules can represent the domain knowledge in the domain knowledge model. An example of the production rules is shown as follows:

If the goal is to solve $A + B = C$ for B

Then rewrite the equation as $B = C - A$

In MT systems, a number of paths for solving a problem are defined and any undefined path is considered as an error. However, incorrect production rules can be

modelled for frequent mistakes. These incorrect production rules are also known as the bug library (Mitrovic et al., 2003; VanLehn et al., 2005). The bug library represents the student's misunderstandings. The MT approach has been used to develop successful ITSs for a variety of domains, including algebra (Koedinger et al., 1997), geometry (Koedinger & Anderson, 1993) and programming in Lisp (Anderson & Reiser, 1985).

CBM is based on the theory of learning from performance errors (Ohlsson, 1996). Ohlsson (1992) suggests that domain knowledge can be represented as a number of constraints. A constraint consists of an ordered pair of $\{C_r, C_s\}$. C_r is the relevance condition and C_s is the satisfaction condition. If, in a scenario, a relevance condition applies, then the satisfaction condition must also be met. For instance, if a driver is driving in New Zealand then s/he must be driving on the left side of the street. When the relevance condition is met and the satisfaction condition has not been met, the constraint is violated. In a constraint-based tutor, the student solution is checked against all constraints in the domain model. If no constraint is violated then the solution is correct; otherwise, the system can provide the student with feedback messages corresponding to each violated constraint. The student model keeps a list of satisfied and violated constraints for all submissions; thus, a constraint-based tutor can track the usage of constraints over time.

In a constraint-based tutor, the student is not restricted to following a specific problem-solving procedure. Therefore, it is possible for the system to support multiple correct solutions. In such systems, it is easy to add new problems (Mitrovic, 2012).

SQL-Tutor³ is the first constraint-based tutor (Mitrovic, 1998; Mitrovic & Ohlsson, 1999). CBM tutors have also been implemented to teach a wide range of domains, including Java programming (Holland et al., 2009), object-oriented software design using UML class diagrams (Baghaei et al., 2007), capital investment (Mitrovic et al., 2008), electronics (Billingsley et al., 2004), English language learning (Menzel, 2006), discrete mathematics (Billingsley & Robinson, 2005), managing oil palm plantations (Amalathas et al., 2012), Newtonian physics (Mills & Dalgarno, 2007) and thermodynamics (Mitrovic et al., 2011).

³ SQL-Tutor is available at <http://ictg.cosc.canterbury.ac.nz:8000>

Example-tracing approach unlike model-tracing and constraint-based methods evaluates student behaviour by comparing it against generalised examples of problem-solving behaviour. Example-tracing tutors provide step-by-step feedback by interpreting a student's problem-solving behaviour with respect to a behaviour graph. The behaviour graph is the generalised solution for the given problem (Alevan et al., 2009). Example-tracing approach has been used for developing different ITSs in different domains such as Stoichiometry (McLaren et al., 2006).

2.4 SQL-Tutor

SQL-Tutor is a web-based, constraint-based tutor that teaches SQL (Structured Query Language) (Mitrovic, 1998; 2003) and is developed and maintained by the ICTG⁴ at the University of Canterbury. In this system, students are given a customised problem-solving environment in the SQL domain. SQL contains three components: Data Manipulation Language (DML) for changing data within a database, Data Definition Language (DDL) for manipulating database objects, and View Definition Language (VDL) for defining views. SQL is a structured language that conforms to set specifications and rules although there could be more than one solution for a problem. 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 300 problems defined on 13 databases. Figure 2.3 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.

⁴ ICTG web page: <http://ictg.canterbury.ac.nz>

SQL-TUTOR Change Database New Problem History Student Model Run Query Help Log Out

Problem 275
 For each author, give the author's name and number of books the author has written. Assign an alias to the total number of books. Order the list in descending order by author's last name.

Check whether you should have ascending or descending order in the ORDER BY clause!

SELECT lname, fname, count(*) as BOOKS_WRITTEN

FROM author, written_by

WHERE authorid=author

GROUP BY lname, fname

HAVING

ORDER BY lname

Feedback Level: Hint

Schema for the BOOKS Database

The general description of the database is available [here](#). Clicking on the name of a table brings up the table details. Primary keys in the attribute list are underlined, foreign keys are in *italics*.

Table Name	Attribute List
AUTHOR	<u>authorid</u> lname fname
PUBLISHER	code name city
BOOK	code title <i>publisher</i> type price paperback
WRITTEN_BY	<i>book</i> <i>author</i> sequence
INVENTORY	<i>book</i> quantity

Figure 2.3 Problem-solving environment in SQL-Tutor

Before submitting the solution to be checked, the student can select the level of feedback they want to receive in case their answer is incorrect (see Figure 2.4). The level of feedback defines how much assistance will be provided to the student. SQL-Tutor offers six levels of feedback: simple (positive/negative) feedback, error flag, hint, all errors, partial solution and complete solution. *Positive/negative feedback* has the lowest level of assistance, and its message informs students whether their answer is correct or not. The message also shows how many errors students have in their solution. An *error flag* message advises students about the clause in which the error happened. More information about the type of error will be provided when a *hint-type feedback* is requested. *Partial solution* message shows a correct content of a clause in the question. A feedback of type *all errors* displays the hint-type messages for all errors the student has made.

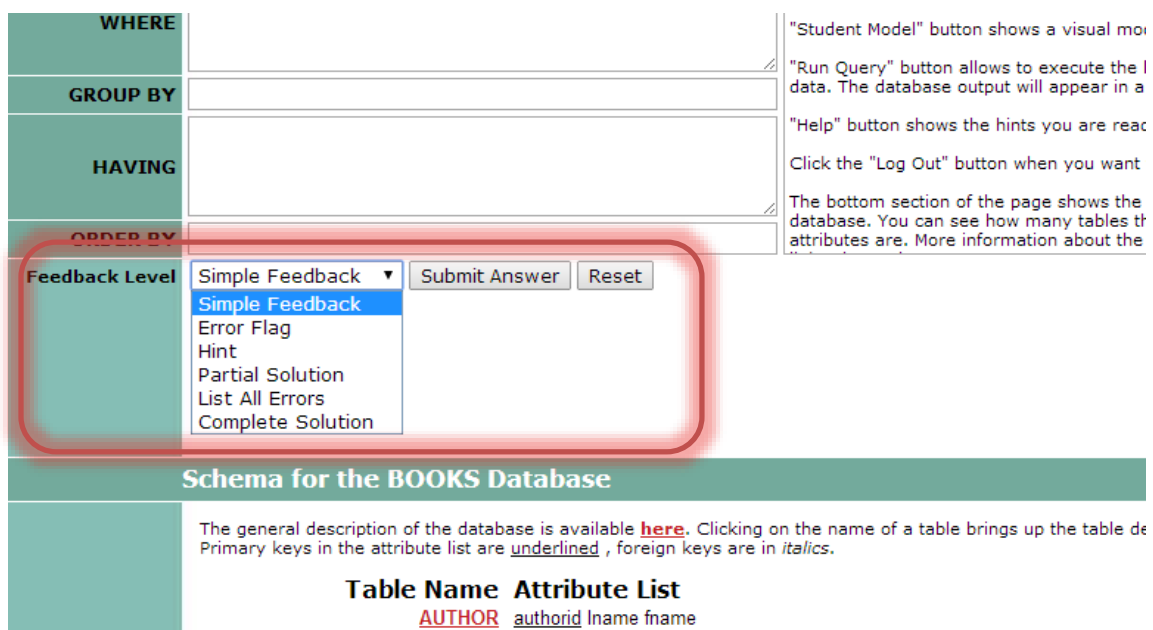


Figure 2.4 Screenshot of feedback levels in SQL-Tutor

At the maximum level, the *complete solution* simply reveals the pre-specified ideal solution of the problem. When a student starts solving a new problem, the default feedback level is *positive/negative* type. As the student goes through several incorrect unsuccessful attempts, the feedback is moved up to the *error flag* and *hint* levels. The system never upgrades feedback to higher than a *hint* type, but students can request any level of feedback at any time while they solve a problem. Students can attempt (submit a solution) many times (Mitrovic & Martin, 2000).

The student may ask for the next problem and the system selects the most appropriate problem for the student based on their student model (i.e. student current knowledge and performance are saved in the student model using violated constraints and problems they attempted). Mitrovic and Martin (2003), Mitrovic and Martin (2000), and Mitrovic and Martin (2004) show various strategies of problem selection and feedback that have been evaluated within SQL-Tutor. Students have access to all their own submitted solutions through the ‘History’ button and can run any runnable query using the ‘Run Query’ button. Moreover, students can get help on how to use the system and change the database at any time during problem solving. An Open Student Model (OSM) (i.e. page shows the student knowledge level) is displayed in both graphical and textual form by clicking on the ‘Student Model’ button. OSM is also known as the Open Learner Model. Figure 2.5 shows a screenshot of OSM in SQL-Tutor.

In Figure 2.5, the system shows the amounts of student understanding of each domain concept. It also shows a relative amount of each concept that has not been covered yet. Therefore, the system suggests the best concept to work on based on student knowledge. See (Mitrovic & Martin, 2007) for more information on OSM.

The architecture of SQL-Tutor is similar to the general architecture we discussed in the previous section. Figure 2.6 shows the architecture of the web-based SQL-Tutor. The databases, problems and ideal solutions are saved in the domain module. Over 700 constraints were used to model the SQL domain, each constraint taking over an hour to develop (Mitrovic, 1998) and has up to two feedback messages (Mitrovic, 2003). The system shows the feedback of violated constraints depending on the pedagogical strategy chosen. All the constraints are problem independent (Mitrovic, 2003).

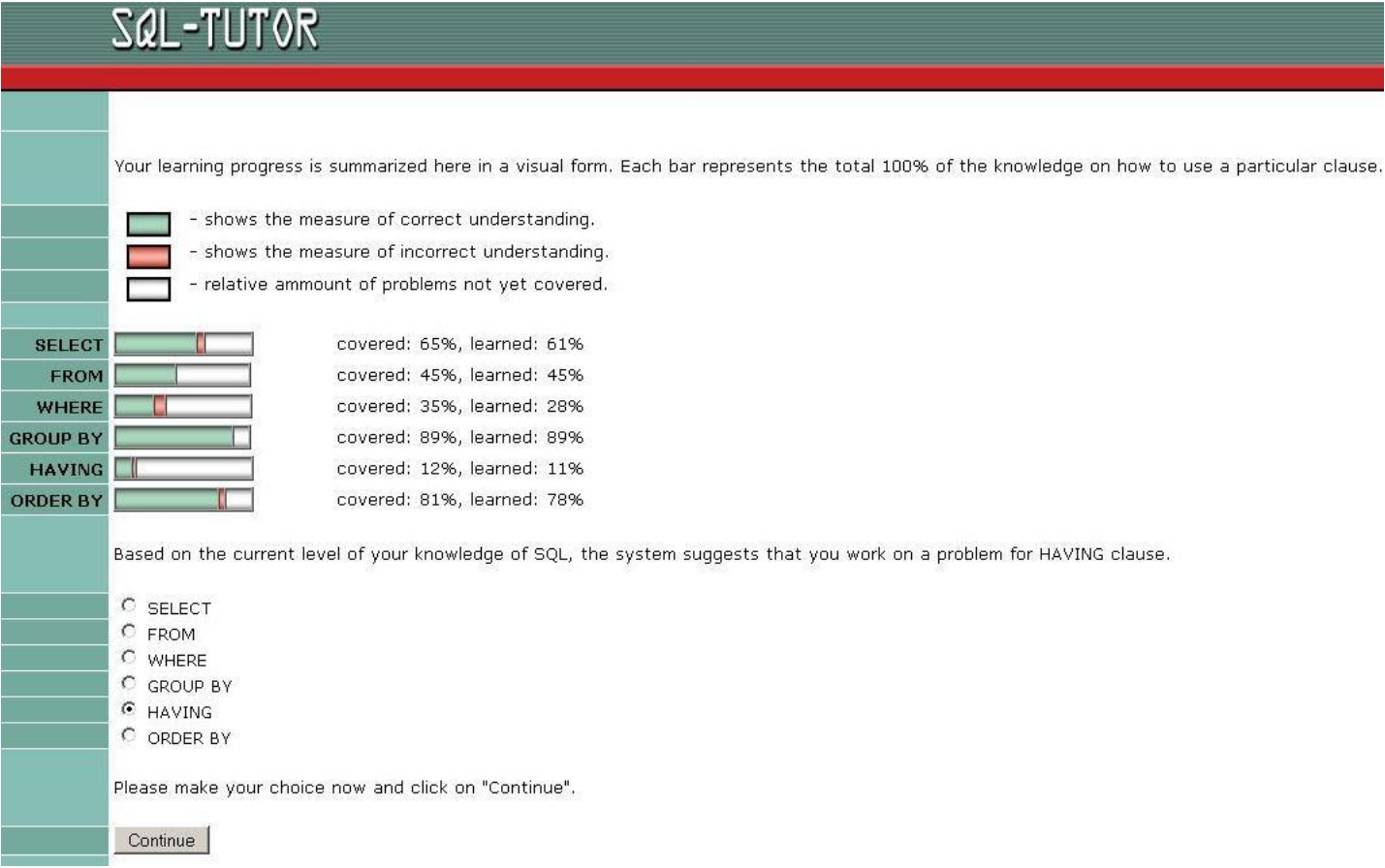


Figure 2.5 Screenshot of Open Student Model in SQL-Tutor

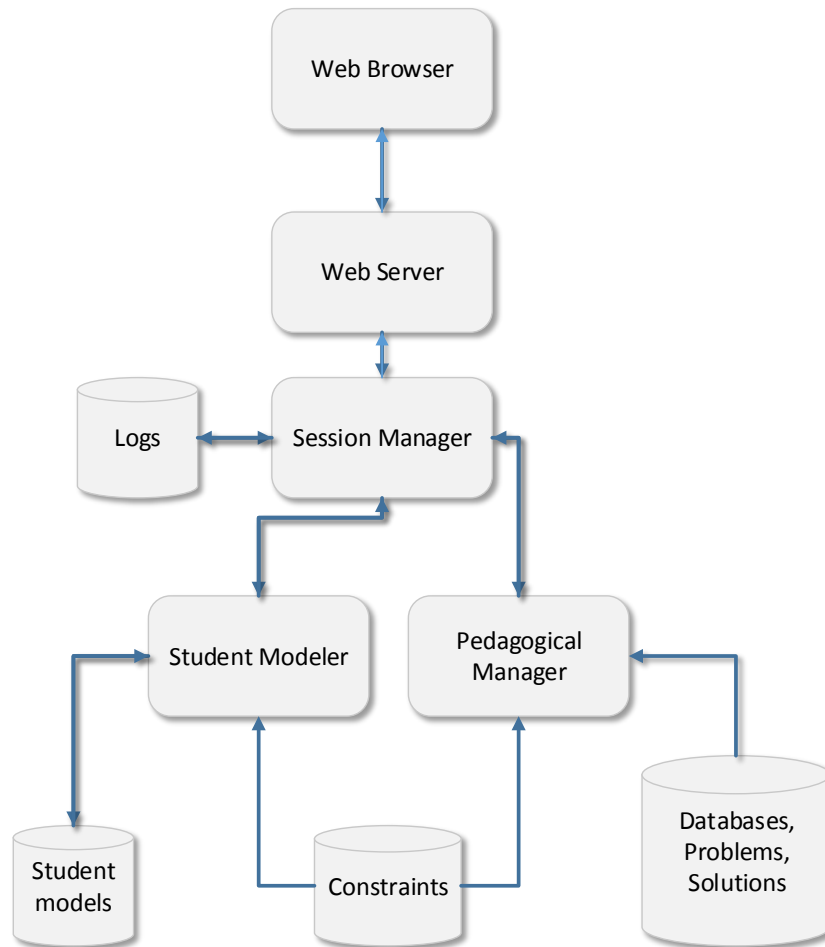


Figure 2.6 The architecture of web-based SQL-Tutor (Mitrovic, 2003)

In SQL-Tutor, constraints identify the syntax and semantics of the domain. Two typical constraints in SQL-Tutor are shown in Figure 2.7. Both constraints are about specifying the join condition in the FROM clause. The first line of each constraint indicates the constraint's number, followed by a hint message which will be shown when the constraint is violated. Constraint 110 is relevant if the JOIN keyword is used in the FROM clause and the satisfaction condition checks whether or not the ON keyword is also used in the same clause. That is, students have to use both keywords (JOIN and ON) to specify a join condition in the FROM clause. Although Constraint 358 is also about the join condition specified in the FROM clause, it has a more specific relevance condition. In case the learner has used both the JOIN keyword and the ON keyword in the FROM clause, the satisfaction condition checks the order of keywords and other elements that have been specified, such as variables that stand for table and attribute names (Mitrovic, 2003).

(p 110

"You need the ON keyword in FROM!"

```
(member "JOIN" (from-clause ss) :test 'equal) ;Relevance condition
```

;ss is the student's solution

```
(member "ON" (from-clause ss) :test 'equal) ;Satisfaction condition
```

```
"FROM")
```

(p 358

"Check the syntax for the JOIN and ON keywords in FROM!"

```
(and (member "JOIN" (from-clause ss) :test 'equalp) ; Relevance condition  
      (member "ON" (from-clause ss) :test 'equalp)); Satisfaction condition
```

```
(match '(?*d1 ?t1 ??s1 "JOIN" ?t2 ??s2 "ON" ?a1 "=" ?a2 ?*d2)  
        (from-clause ss) bindings)
```

```
"FROM")
```

Figure 2.7 Two constraints that checks the syntax (Mitrovic, 2003)

Constraints 110 and 358 check the syntax error while Figure 2.8 shows a typical constraint that also checks the semantics. Constraint 11 is relevant for the solutions that contain JOIN, ON and the corresponding variables in the FROM clause, where two variables are from the tables in the currently selected database and one of the remaining variables is an attribute of the first table. In such cases, the satisfaction condition states that if the other attribute specified in FROM is the attribute of the second table and the types of the attributes are the same, then the solution is correct. For more examples see (Mitrovic, 1998; 2003; Mitrovic & Ohlsson, 1999).

The name of the clause that a constraint refers to is shown in the last part of each constraint. For instance, in Figures 2.7 and 2.8, the three constraints deal with the FROM clause (Mitrovic, 2003).

Because SQL-Tutor does not have a problem solver module, it is not capable of solving a problem on its own. Alternatively, developers provide the ideal solution for each problem and the system compares the student solution by matching it to constraints and

ideal solutions. Therefore, SQL-Tutor has a number of constraints that check the student solution with the ideal solution (Mitrovic, 2003).

(p 11

"If the JOIN keyword is used in the FROM clause, the same clause should contain a join condition specified on a pair of attributes from corresponding tables being joined."

; Relevance condition

(and (match '(?*d1 ?t1 ??s1 "JOIN" ?t2 ??s2 "ON" ?a1 "=" ?a2 ?*d2) (from-clause ss) bindings)

; FROM contains variable t1, JOIN followed by variable t2, the ON keyword and a comparison

(valid-table (find-schema (current-database *student*)) ?t1)

; t1 is a table from the current database

(valid-table (find-schema (current-database *student*)) ?t2)

; t2 is a table from the current database

(attribute-of (find-table ?t1 (current-database *student*)) ?a1))

; a1 is an attribute of table t1

; Satisfaction condition

(and (attribute-of (find-table ?t2 (current-database *student*)) ?a2)

; a1 is an attribute of table t2

(equalp (find-type ?a1) (find-type ?a2)))

; a1 and a2 must be attributes of the same type

"FROM")

Figure 2.8 Constraint that checks the semantics (Mitrovic, 2003)

SQL-Tutor is run on the AllegroServe⁵ web server and students can access the system through the internet. See (Mitrovic, 2003) for a more thorough explanation of SQL-Tutor.

2.5 Self-Explanation (SE)

Self-explanation is a metacognitive activity in which a student explains the provided learning material to him/herself. Research shows that students who explain learning material to themselves learn more than students who receive explanations (Chi et al., 1994; Brown & Kane, 1988; Webb, 1989; Hattie, 2009). Very few students self-explain spontaneously, but can be encouraged to self-explain with carefully designed prompts (Chi et al., 1989; Chi et al., 1994).

⁵ Is available from Franz Inc: <http://www.franz.com>

Prior studies, either with a human teacher prompting self-explanation (Chi et al., 1994) or with an ITS prompting self-explanation (Weerasinghe et al., 2009; Weerasinghe et al., 2011) show that self-explanation is an effective metacognitive strategy. Alevan and Koedinger (2002) investigated the effectiveness of self-explanation of individual steps in problem solving by prompting students to select the underlying principle within the Geometry Cognitive Tutor, and showed that students who engage in self-explanation acquire deeper knowledge.

Chapter 3. Using examples in learning

Many learners prefer to see a worked-out example (i.e. a problem with the solution) before attempting to solve a problem. This happens more when the learner has a low prior knowledge. Sweller (2006) explains the *worked-example effect* based on the Cognitive Load Theory (CLT). He shows that examples decrease cognitive load on the learner's working memory; thereby, the example-based strategy is more helpful for novices who have to deal with an enormous amount of cognitive load. In the following sections, we look at previous studies from two different angles: studies comparing examples with unsupported problem solving and studies comparing examples with supported problem solving in Intelligent Tutoring Systems. In unsupported problem solving, learners do not get any feedback, while in tutored problem solving students receive feedback on their solution steps and final answers. Before we explain these studies, first we must know what an example is and how it works. Thereby, we begin with studies that define the mechanism of learning from examples.

3.1 Cognitive Load Theory (CLT)

In 1956, George Miller introduced “the magical number seven, plus or minus two” to the world, which indicates that our cognitive system can only process 7 ± 2 items at any time (Clark et al., 2006). Cognitive load theory uses our knowledge of human cognitive architecture to help with instructional design, which results in more efficient learning. Clark et al. (2006, p. 7) indicate that “Cognitive load theory is a universal set of learning principles that are proven to result in efficient instructional environments as a consequence of leveraging human cognitive learning processes”. The remaining question is how examples can improve learning.

Sweller (2006) presents different cognitive load theory principles and their relation to the worked-example effect. A worked example is a written step-by-step instruction on how to solve a problem, and the worked-example effect has been explained in terms of the different amounts of cognitive load evoked by studying examples and solving problems. In the *borrowing principle*, the learner borrows the needed information from other people's long-term memory and connects the new information with the prior

knowledge. The learner can then assign the meaning to the new information. Therefore, Sweller believes that worked examples are the representation of the borrowing. On the other hand, in the *randomness as genesis principle*, new information is created based on the prior knowledge in the learner's long-term memory. This process requires large amounts of information in working memory. Problem solving is the ultimate instantiation of this principle. In addition, the randomness principle is normally used when applying the borrowing principle is impossible.

There are three different loads for the working memory: intrinsic load, extraneous load and germane load. *Intrinsic load* is caused by the nature of concepts such as the difficulty of tasks, so in as much as a problem is more complex, its intrinsic load is higher. Although the inherent intrinsic load of an instructional content cannot be directly altered, it is possible to manage the intrinsic load of any lesson by breaking down complex tasks into a number of smaller tasks with lower intrinsic load. For instance, instructional professionals decompose complex tasks into a series of prerequisite tasks when they outline their courses and lessons (Clark et al., 2006). *Extraneous load* is the information not related to learning, such as noise in class. In contrast to extraneous load, *germane load* is the information which is related to learning materials. Clark et al. (2006) outline different strategies and instructions to decrease extraneous load and increase germane load.

To summarise, worked examples reduce the time our cognitive systems need to bring previously learnt material from the long-term memory into the working memory. Moreover, if the learner needs specific items to understand a new concept and the needed items are not in their long-term memory (lack of prior knowledge), then worked examples can provide the items for the learner. Another explanation for the advantages of examples against other learning methods is that examples decrease the extraneous load on working memory (Clark et al., 2006), while problem solving for instance imposes a high extraneous load.

Cognitive load research has shown that worked examples help students to learn more efficiently (faster) compared to problem solving, yet it is not completely clear whether or not they learn more effectively (gain more) with worked examples.

3.2 Worked examples and modelling examples

Worked examples provide worked-out solution steps that need to be followed to reach the goal (Sweller & Cooper, 1985). In addition, for teaching complex problem-solving skills, studying worked examples is a suitable instructional strategy (van Merriënboer, 1997).

In modelling examples, the problem solution is demonstrated to learners by a model (the model can be an animated agent or a human). The main difference between worked examples and modelling examples is that worked examples usually show a written ideal solution, while in modelling examples learners watch to see how another person solves the problem. There are a number of design characteristics that are not important in worked examples but are relevant for modelling examples (e.g. sex, age, competence, background or model-observer similarity) (van Gog & Rummel, 2010). For instance, low achieving students might learn more effectively from non-experts and high achievers from experts, because models with low expertise might make a mistake or an expert model might skip some solution parts (Kalyuga & Sweller, 2004).

3.3 Self-explanation effects in the example-based strategy

CLT explains that examples reduce extraneous load on working memory (Sweller et al., 2011). Extraneous load is caused by the way in which learning material is presented and does not directly contribute to learning (Clark et al., 2006). By reducing extraneous load, a part of working memory becomes available. If we load that freed working memory with germane load, it improves learning. One way of producing germane load is to prompt students to self-explain. In a study to teach concept mapping, Hilbert and Renkl (2009) showed that students who self-explained after they studied examples learnt more than students who did not engage in self-explanation. In another study, Schworm and Renkl (2006) found that self-explanation is effective for studying worked examples and solved-example problems. Solved-example problems only provide the problem formulation and the solution, while worked-out examples consist of a formulation, solution steps and the final answer.

We believe that if self-explanation (SE) prompts are designed specifically to complement problem solving and learning from examples, learning will be more

effective. Previous research has shown that students who studied examples acquired more conceptual knowledge than procedural knowledge, and students who solved problems learnt more procedural knowledge than conceptual knowledge (Kim et al., 2007; Schwonke et al., 2009). This suggests that different types of SE are needed to scaffold problem solving and examples.

SE prompts can be of different natures, according to the knowledge they focus on. For instance, Hausmann et al. (2009) 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 (1991) 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?”).

To summarise, SE has an effective impact on learning from examples and problems. However, learning from examples should be complemented with SE prompts focused on procedural knowledge, while problem solving should be complemented with SE prompts about the relevant domain principles.

3.4 Learning from examples vs. unsupported problem solving

There has been no agreement on how much assistance should be provided to students. Sweller et al. (2011) show that maximum assistance (e.g. example) is more efficient than minimal assistance when first learning in a new domain (e.g. unsupported problem solving) which has been corroborated by prior studies such as (Atkinson et al., 2000). Apart from the advantages of learning from examples versus unsupported problem solving, recently researchers have focused on different example-based learning strategies. Van Gog et al. (2011) investigated 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 (problem solving was unsupported). The tasks were from the domain of electrical circuits troubleshooting and the experiment ran in four sessions. First, some general information was given to

participants about the experiment, followed by a pre-test. Then participants started the condition-associated training tasks. They used a self-reported mental scale to measure the actual cognitive load for each task. Participants solved two problems (post-test) after the training task. The experiment was controlled for time. The results show that the participants in WE and WE-PS had higher performances in the post-test than PS and PS-WE. Furthermore, mental effort training and test rates in WE-PS and WE were lower than PS and PS-WE.

In a later study, van Gog (2011) stated that the previous results on WE-PS and PS-WE might not be sufficient. Examples which come after problems had a different structure to the next problem; therefore, she opined that using identical pairs might lead to a different result. She conducted a study using modelling examples (ME) and problem solving (PS) in two conditions (PS-ME-PS-ME and ME-PS-ME-PS) in the Leap Frog game. In modelling examples, the problem solution is demonstrated to learners by a model (van Gog & Rummel, 2010). After two sequences of training, students worked on two tasks, of which the second was not similar to the training tasks. There was no difference in learning performance since the students learnt the most after studying the second worked example.

In another study, student's prior knowledge had an important influence on instructional formats. Formats which are efficient for some students might not be efficient for a student with a different knowledge level (Kalyuga, 2007). In other words, if the additional information was not needed by the student, the *expertise reversal effect* was observed (Kalyuga et al., 2001). Expertise reversal effect is caused by redundant information. Information that is essential for novices could become redundant as the level of expertise increases. Redundancy effect happens when different sources of information describe the same concept. Eliminating redundant information reduces the cognitive load (Chandler & Sweller, 1991). Therefore, analysing a worked example under such conditions might cause unnecessary cognitive load that interferes with learning (Kalyuga et al., 2001).

Most of the prior studies showed the worked-example effect in well-defined problems. Well-defined tasks are those for which there exist algorithms for solving problems (Mitrovic & Weerasinghe, 2009), such as mathematics and physics. Rourke and

Sweller (2009) present two studies in which they investigated the worked-example effect using ill-defined problems. They hypothesised that students who learnt to recognise a designer's work from observing examples of that designer's work could recognise other work of the same designer more easily than students who learnt from solving equivalent problems. Both studies consisted of three phases. In the first phase, students participated in a design history lecture and afterwards studied a worked example and solved a problem or two problems, according to the condition they were in. In the last phase, students answered visual recognition and short answer tests. The difference between the studies was in the participants' abilities. Students in the second study had a greater level of visual literacy skill than the students in the first study, although both studies' participants had the same knowledge level on design history. The results of both studies show that the worked-example effect can be obtained in ill-defined as well as well-defined domains.

Kyun et al. (2013) investigate the worked-example effect in another ill-defined domain. Participants were university students enrolled in the "Children's English Literature" course. The students were asked to write essays. Although English was the participants' second language, they had sufficient language proficiency to write and discuss their ideas in English. They conducted three studies. Each study had two conditions: worked examples and problem solving. In Experiment 1, the learning phase for the problem-solving condition involved writing essays for two similar questions, but participants in the worked-examples condition saw possible answers to the first question, then wrote an essay for the second question. Researchers found a significant difference in cognitive efficiency between the two groups, with the worked-examples condition superior to the problem-solving condition. Since the participants in Experiment 1 had high levels of knowledge in literature, researchers argue that the expertise reversal might have influenced the result. Therefore, they conducted two similar experiments using participants with less literature knowledge. Participants in Experiment 3 had the lowest expertise level in comparison with Experiments 1 and 2. Results in Experiment 2 indicated significant differences between the two learning strategies in cognitive efficiency and retention tests. In Experiment 3, all the results showed significant differences in favour of the worked-examples condition. Overall, the paper shows that the worked-example effect increases when the level of expertise decreases in an ill-defined domain such as writing essays in English literature.

Kalyuga (2009) reviewed a number of prior studies and proposed five instructional supports to enhance learners' ability to transfer their knowledge: faded worked examples, worked examples and tutored problems, worked examples and self-explanation, worked examples and visual mapping, and worked examples plus self-visualisation. In addition, Hilbert and Renkl (2009) investigated the best structure of examples to teach concept mapping. They found that students learn more when examples are presented with self-explanation than without it.

In the previous section, we mentioned that worked examples reduce the extraneous load. Therefore, the freed memory should be used by germane load. SE prompts induce germane load into working memory, consequently improving learning. Providing SE prompts is one way to scaffold worked examples.

The other approach to scaffold worked examples is to give a testing task after students study an example. Roediger and Karpicke (2006) show that after studying material, testing is more effective for long-term retention than restudying; this is called the 'testing effect'. Van Gog and Kester (2012) investigate the testing effect when acquiring problem-solving skills. They discussed studies that showed the advantage of a testing task after studying an example in retention. This study is the first to compare testing effects when students acquire problem-solving skills. Van Gog and Kester (2012) created two conditions: SSSS and STST (S: study example, T: testing task). Each condition had two pairs of tasks (SS or ST) and the tasks in each pair were isomorphic. They developed trouble-shooting problems from the 'electrical circuits' domain. Examples were similar to testing tasks with solutions. They conducted a conceptual pre-test. Next, students had 3 minutes for each task and could not return to an example or problem passed. Five minutes after finishing the learning phase, students were given an immediate post-test consisting of two trouble-shooting problems. Finally, students took a similar post-test after one week. The results showed no significant difference in the immediate post-test. In the delayed post-test, students who only studied examples outperformed their counterparts from the other testing conditions. That is, the testing effect might not apply to acquiring problem-solving skills from worked examples.

This is a surprising result and in contrast with prior findings. They explained their result with three possible reasons. The main difference between this study and prior

research was that students had to construct the answer in addition to recalling from memory. Therefore, ‘answer construction’ may interfere in the recall process. Another reason is the study duration, which might be too short. The next possible reason is that examples induce self-explaining and self-explanation correlates with a longer retention; thus, SSSS performed better than STST in the delayed post-test.

Although research shows that students learn more effectively from worked examples than solving unsupported problem solving, worked examples have their own drawbacks. First of all, examples must be studied to be effective; thus, when an example is ignored, it does not promote learning. Some students either skip the worked examples completely, or pay little attention while studying worked examples. On the other hand, problems require deep processing for solution. Therefore, replacing worked examples with partly worked examples is one way to minimise ignoring worked examples (Clark et al., 2006). Partly worked examples are also known as faded examples (i.e. a worked example in which one or more steps are left for the student to complete). Faded worked examples must be superior to conventional problems, as partly worked examples require less effort during training and impose less cognitive load than a conventional problem to solve.

Paas (1992) reports the results of a study in which worked examples, faded examples and conventional unsupported problems were compared. The materials were from the domain of basic statistics. The study had three conditions: unsupported problems, worked examples and problems pairs, and faded examples and problems pairs. The results showed that faded examples and worked examples were superior to conventional unsupported problem solving for attaining transfer. Learning from worked examples and faded examples was the same.

In this section we presented a review of research that compared learning from worked examples to unsupported problem solving. In the next section, we look at the worked example compared to tutored (supported) problem solving.

3.5 Learning from examples vs. tutored problem solving

Many prior studies have compared worked examples to unsupported problem solving. Koedinger and Alevan (2007) criticised these prior studies because of the very different

amounts of information provided in the two conditions (the unsupported problem-solving condition received no feedback upon submitting a solution).

As the response to this criticism, Schwonke et al. (2009) compared a standard cognitive tutor (Geometry Tutor) with a new version that was enriched with faded worked examples. They conducted two experiments. In the first, students in the problem-solving condition worked with pure problem-solving tasks and students in the example condition worked with fixed faded examples. The result revealed an improvement in learning time from using examples. In the second experiment, they used the think-aloud protocol in order to study relevant cognitive processes. According to the result, the efficiency advantage of worked examples was replicated. Salden et al. (2010) reviewed a number of prior studies on worked examples (e.g. McLaren et al., 2008; Anthony et al., 2008) and concluded that using worked examples in tutored problem solving decreases learning time.

To summarise, using worked examples is efficient in ITSs, especially for novices since they do not have adequate prior knowledge and examples provide the needed information. Therefore, it could be assumed that using a combination of examples and problem solving might lead to a better result.

McLaren and colleagues (2008) discussed three studies they conducted on example-based strategy using the stoichiometry tutor in which they compared tutored problem solving to learning from worked examples combined with tutored problems. The students in the problems condition worked on solving tutored problems, while students in the examples condition observed worked examples, were prompted to self-explain, and solved isomorphic tutored problems. Both groups had to take pre- and post-tests. In all three studies, students in the examples condition learnt faster, but there were no significant differences in the near transfer (far transfer was not measured). Authors suggested that one possible reason for no difference in knowledge acquisition is that students in the problems condition made initial problems into worked examples by clicking on the hint button, then tried to solve the next isomorphic problems. From the authors' point of view, that might be the reason for the same near transfer for both groups, with the worked-examples condition taking much less time to complete.

In a recent study, McLaren and Isotani (2011) compared examples only, alternating worked examples/tutored problems and all tutored problems. They conducted the study using the Stoichiometry Tutor and modelling examples. Surprisingly, the result showed that students benefitted most from learning with worked examples only, at least with respect to learning time. However, examples were followed by self-explanation prompts while the problems were not. They also discovered that using interactive worked examples may sometimes be more beneficial than static worked examples and tutored problems. In interactive worked examples, students were asked about their understanding of the examples (e.g. comprehension questions).

Corbett et al., (2010) investigated the interleaving examples with an ITS and learning activity that focuses on domain knowledge (process modelling). They had four conditions: process modelling and problems (MOD), examples then problems and process modelling (ALL), examples and problems (IWE), and problem solving only (PS). They found no difference between the conditions on the robust learning and problem-solving tests. Process modelling led to greater accuracy and faster reasoning. Faster reasoning in problem solving can be achieved by using both worked examples and process modelling together. Students in all the conditions learnt the same problem-solving knowledge which was measured by the post-test accuracy.

Although the aforementioned studies show the advantage of using examples in ITSs, more recent research yields opposite results. Corbett et al. (2013) report on three empirical studies. The studies evaluate the impact of interleaved worked examples and genetic-process reasoning scaffolds in an ITS. For the genetic-process reasoning, they developed two types of tasks: process modelling tasks and solution construction tasks. The tasks were designed to precede standard genetics problem-solving tasks and ground problem-solving knowledge of students in the underlying genetics before problem solving. These two learning activities were developed for three topics in the context of genetic problem solving and the ITS they used was the Genetics Cognitive Tutor. Each study included three conditions: a standard problem-solving condition, a scaffolded reasoning condition and an interleaved worked-examples condition. In the interleaved worked-examples condition, students worked with a set of problems, in which worked examples were interleaved with standard problems. In the scaffolded reasoning

conditions, students had a block of scaffolded reasoning problems; the block being designed to prepare students for solving a problem. They found that interleaved worked examples resulted in less basic-skill learning than problem solving. Moreover, the scaffolded reasoning yielded more robust understanding than problem solving. Corbett et al.'s (2013) findings corroborate Salden et al.'s (2009b) study, which showed that incorporating worked examples into an ITS was not superior to the ITS alone.

Overall, most of the studies show that using worked examples in ITSs results in reduced learning time. Although there are some studies showing higher transfer performance for faded examples, most studies have found no differences in the amount learnt. We think that using adaptive worked examples might be more helpful if it is reinforced with a problem-solving approach.

In addition, all prior studies using examples in ITSs were in Geometry, Chemistry and Algebra domains. All these tutors teach well-defined tasks. For ill-defined tasks there is no specific procedure to solve a problem; thus, examples do not reveal essential procedures to find solutions. Although in some cases there might be more than one solution for a well-defined task, most ill-defined tasks have multiple correct solutions (Mitrovic and Weerasinghe, 2009). There is a need for more research in this area in order to explore the usage of examples in ill-defined tasks. Rourke and Sweller (2009) show that the worked-example 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.

3.6 Related work on adaptive worked examples

Using examples improves learning gain of novices by freeing up working memory capacity. However, worked examples become detrimental when students gain more expertise. Then, students learn more from solving problems than studying examples (Kalyuga et al., 2001). Clark et al. (2006) suggest the process of backwards fading of worked examples, which is worked examples transitioning into problems as students gain more expertise. If the process of backward fading becomes adapted to the student's expertise, a higher learning gain can result. None of the aforementioned studies used adaptive examples in comparison with tutored or unsupported problems solving.

Salden et al. (2009a) compared fixed faded worked-out examples with adaptive ones. Fixed faded examples were the same for all students, but the solution steps in adaptive faded examples were faded based on the student's prior knowledge. They conducted two studies, one in a lab (in Germany) and the other in a classroom (in Pittsburgh). Their main hypothesis was to see whether using adaptive examples combined with problem solving compared to pure problem solving could lead to better learning. They tested three conditions: traditional problem-solving cognitive tutor, cognitive tutor enriched with fixed examples, and cognitive tutor enriched with adaptive faded examples. The lab results indicated that adaptive examples led to better learning and higher transfer compared to the other conditions. In contrast, the classroom results depicted no significant difference in the immediate post-test, but in the delayed post-test students who used adaptive examples learnt more. They believe that the difference in the lab and classroom results might have been caused by either inherent noise in the class compared to the lab, or by not using Cognitive Tutor's mastery criterion in the class. The mastery criterion option led students to enjoy remedial problems for the concept they had not mastered yet until all learning materials were learnt. Therefore, each student completed slightly different sets of problems.

Kalyuga and Sweller (2005) proposed an adaptive model for using examples. Their model works based on learners' Cognitive Efficiency (CE)⁶, which is calculated from students' performance and the cognitive load scores. Students indicated the cognitive load after they complete a task. Kalyuga and Sweller used a different formula from what was previously proposed (van Gog and Paas, 2008; Paas and van Merriënboer, 1993) as they needed to calculate CE in real time during the experiment. Performance was measured based on the number of steps the student required to solve a problem. For instance, when the problem is $2(3x - 1) = 1$, some students start with $2*3x - 2*1 = 1$, thus requiring more steps to solve the problem than students who start with $6x - 2 = 1$ or those who provide the final answer ($x = 0.5$) immediately. The method was tested using the Algebra cognitive tutor enriched with worked examples and faded examples. Students worked with two versions of the tutor: adaptive and non-adaptive. Students in the adaptive condition were allocated to one of the four stages of faded examples (stage 1 fully

⁶ Cognitive efficiency = Performance / Cognitive Load

worked-out examples, stage 4 fully problem-solving tasks) based on their cognitive efficiency scores in the pre-test. All students had to proceed to the final stage of fading (stage 4) from the stage they started. In each stage, a diagnostic task decided if the student needed more information (in the forms of two worked examples or four shortened worked examples). Thirty students were randomly assigned to 15 pairs of students. One student in each pair worked with the adaptive version and the other worked with the non-adaptive version. The student in the non-adaptive condition started from the faded stage in which the other participant in the adaptive condition started. The results showed that students in the adaptive condition scored marginally significantly higher than students in the non-adaptive condition. Students in the adaptive condition showed significantly higher efficiency gains than students in the non-adaptive condition.

In the study performed by Kalyuga and Sweller (2005), the system asked students to indicate how difficult the task was for them. However, this measure is not appropriate for cognitive load (van Gog & Paas, 2008). A student may find a problem very difficult and not invest enough effort to solve it by requesting a complete solution. Instead, van Gog and Paas (2008) suggest asking students to indicate how much effort they have invested into solving the problem.

3.7 Is using examples always helpful?

As far as we know, Corbett et al. (2013) show for the first time that using worked examples is less beneficial than solving tutored problems only. The following research corroborates Corbett et al.'s (2013) study where worked examples were compared with unsupported problem solving.

Kalyuga et al. (2001) conducted two experiments in the context of programming a programmable logic controller (PLC). The results showed that when students had low prior knowledge, they benefitted more from studying examples than from problems to solve. Then worked examples became redundant, with more experience in the domain. Thus, problem solving proved superior to worked examples for advanced students.

Moreno (2006) reviewed a number of empirical studies that could not prove the worked-example effect. She believes that students might not have learnt from examples,

because they could not find relevant information in the examples. Students often suffer from an *illusion of understanding* and as a result they may not be interested in engaging in SE activities (Gerjets et al., 2006). Therefore, it is difficult for students to generalise solutions from examples to novel problems.

Moreno (2006) reviews a few studies that used different methods to improve learning gain using Cognitive Load Theory. The first method looks at ways that can decrease the extraneous load in learning from worked examples. In the second method, researchers investigate different approaches to increase the germane load in a worked example's instruction. Finally, they try to decrease intrinsic load within a worked example's instruction. Moreno believes that the prior studies on these three methods were not satisfying, as they did not measure intrinsic, germane and extraneous load.

The recent effort in optimising examples did not significantly improve learning. Moreno thinks that there is no more effect because there is no way to optimise more, or, as the American saying goes, "We can't squeeze blood out of a turnip". Therefore, she suggests changing the focus of research to the techniques aimed at improving far transfer.

According to Moreno (2006), there is considerable evidence that worked examples do not always improve learning, and the cognitive load field is unable to explain the reasons behind that. In addition, researchers argue that example-based learning is dependent on different factors, such as design of examples and individual differences in example processing (related to students' SE). Moreno suggests extending this list to three factors: worked-example design, individual differences, and essential processing amount during learning. Moreover, worked-example design consists of nine different factors that are shown as follows:

- Different intra examples design
- Less or more information present in the solution steps
- One or multiple solutions
- Sub-goal highlighting
- More or less integrated verbal and visual representation of the problem solutions
- Mixed modality representations of problem solutions
- Different inter-example designs

- More or less variability of surface and structure features
- Different sequencing of worked examples and practice problems

The effectiveness of examples does not depend on example design only, but also on the kind of activity prompted by worked-example methods. Then, examples can be presented in one of the following ways: as using SE prompts, presenting faded examples (incomplete examples), using mixed-initiative problem solving, or providing learners with sub goals.

Moreno (2006) says that the learning benefits of example-based strategy which encourage students to get involved in essential cognitive processing are not only dependent on the level of the student's prior knowledge. For instance, media and domains can be more interesting for some students than others; hence, they will be more motivated to be involved in self-regulation and other techniques.

3.8 Related work on using examples in the SQL domain

There have not been any studies on the effect of examples in SQL-Tutor. However, Mathews and Mitrovic (2009) used examples in SQL-Tutor for their research on framing. Their study consisted of two groups. The first group (experimental) enjoyed a pre-action (priming) phase and post-action (reflection) phase. In the priming phase, the human expert explained most common mistakes and gave some related examples. In the reflection phase, s/he prompted each student to reflect on her or his problem-solving experiences (students were encouraged to analyse their errors and find their misconceptions). This research aimed to evaluate the framing strategy and the results show that the learning time of the experimental group (who get prior knowledge from the expert before working with SQL-Tutor) was shorter than the learning time of the control group. This research focused on the advantages of priming in learning; therefore, although it gave a better view for our future study, it cannot be considered as a complete study on using examples in SQL-Tutor.

In a study by Catrambone and Yuasa (2006), the example-based strategy was compared to unsupported problem solving. They compared instructional elaboration and self-generated elaborations guided by scaffolding in active (problem solving) and passive

(examples) conditions. Therefore, they set-up four considerations: Action Active (CA-ACT), Elaboration Active (C-ACT), Action Passive (CA-PAS) and Elaboration Passive (C-PAS). The study was conducted in three phases. First, learners had to read the SQL manual and the active groups received structured exercises after reading manual. In the second phase, learners were given a criterion test. The aim of this test was to make sure the participants knew the SQL operators and procedures. If the participant could pass all the questions in the second phase correctly (if the participant gave incorrect answers, they could refer to the manual or ask the experimenter for the correct answer), then they were allowed to proceed to the third phase, which was writing SQL queries. In the third phase, participants were tested using a computer-based SQL environment. The system provided feedback based on the common errors. The results showed that, although the active group needed more training time, they did the post-test faster than the other groups. Moreover, students in the elaboration condition learnt slower than the action group. The participants in the elaboration condition needed more hints to carry out the tasks and committed more important errors during the query writing.

Overall, prior studies on the SQL domain have not directly addressed or investigated the worked-example effect versus the tutored problem solving. We think that incorporating examples into SQL-Tutor improves learning; however, this process should be adapted to students' knowledge levels.

3.9 How should examples be delivered

The question is, how to deliver examples to optimise effectiveness? Overall, prior studies have found that in some conditions, examples only and example-problem pairs are more efficient than the problem-example pairs and problems only (e.g. (van Gog et al., 2011; McLaren & Isotani, 2011)).

Gerjets et al., (2006) compare effects of using modular examples versus molar examples in the context of probability. In molar examples, the basic units of analysis are solution procedures and problem categories that cannot be broken down any further. Learning from molar examples can simply fill all the working memory, because they can easily become complex and cause a high intrinsic load. In contrast, modular examples emphasise individual solution steps below the category level. That is, solution steps are

broken down into smaller procedures that can be learnt in isolation and transferred separately when solving novel problems. They also investigated whether or not prompting SE or providing instructional explanation with molar or modular examples may improve learning. The hypothesis was that molar examples-instructional examples pair and modular examples-SE pairs lead to a higher improvement in learning compared to the other possible conditions. The reason behind their proposition is grounded on cognitive load theory. They believe that the increase in intrinsic load caused by molar examples can be paid back by memory freed by instructional explanation. Modular examples have lower intrinsic loads; thus, SE can improve learning with germane load.

In the first experiment, they investigated the effects of providing different levels of instructional explanations (low, medium, high) with modular or molar examples (3*2 conditions). The results revealed that instructional examples were superfluous when the molar examples compared to modular examples. They mentioned some reasons, such as the quality of instructions and split attention as possible reasons for these results. The result was assessed using modified version of the NASA-TLX and included task demands, effort, navigation demands, success feeling and stress. Moreover, they found that learning with the modular approach took almost half the time of learning with the molar approach. Moreover, the students in modular examples retrieved fewer examples, solved more problems correctly, showed lower cognitive load and felt more success than students who worked with molar examples.

In the second experiment, Gerjets et al., (2006) investigated the effects of prompted SE on molar and modular examples. The result again reported the advantages of modular examples in learning time. Furthermore, they found that prompted SE increased the time of learning. They showed that prompting SE was stressful for the students. The result showed that SE prompts did not improve learning when provided with molar and modular examples. They explained the result with different reasoning. For instance, the number of similar SEs might cause the redundancy effect; or providing SE answers on the next page (spatial contiguity principle) could cause the split attention effect.

According to the aforementioned studies, SE has a high influence in learning from examples; it fills up freed intrinsic loads (as a result of using examples) in working memory with germane loads. Therefore, their accompaniments with examples are

inevitable. The problem with SE is that students may overestimate their understanding from the examples due to an illusion of understanding; as a result, they may not be interested in engaging in SE activities (Gerjets et al., 2006).

Another effective delivery strategy is ‘completion’ or ‘fading strategy’; that is, some steps in the example are faded and the learner must complete these omitted steps. Then, in the next example, the number of faded steps will increase. Hence, the learner starts from a complete worked example and continues to problem solving (Renkl & Atkinson, 2003). Renkl and Atkinson (2003) found that the fading strategy even more effective than example-problem pairs and problem-example pairs. However, Reisslein et al. (2006) found no overall difference between them.

The last effective delivery strategy is ‘random sequence’. This strategy suggests that examples should not be presented in a typical blocked sequence by problem type or category (van Gog & Rummel, 2010). Based on CLT, the random sequence increases the cognitive load in working memory, but leads to better learning and transfer outcomes.

Considering previous work in using examples, the best way to deliver examples is when we can adapt them to learners’ characteristics. For instance, using faded examples improves learning gain by considering a student’s prior knowledge. Sex and age might influence learners’ interest in learning from examples or ITSs. This leads us to various types of an example’s adaptability, which have been less investigated compared to ITSs.

Apart from an example’s delivery method, it is important to know how to design an example. Mayer (2002) suggests seven principles in designing multimedia materials:

1. **Multimedia principle:** Students learn better from words and pictures than from words only.
2. **Spatial Contiguity principle:** Students learn better when related words and pictures are presented closer rather than far apart and on different pages and screens.
3. **Temporal Contiguity principle:** Students learn better when corresponding words and pictures are presented simultaneously rather than successively.
4. **Coherence principle:** Students learn better when extraneous words, pictures and sounds are excluded rather than included.

5. **Modality principle:** Students learn more from animation and narration than from animation and on-screen text.
6. **Redundancy principle:** Students learn more from animation and narration than from animation, narration and on-screen text.
7. **Individual differences principle:** Design effects are stronger for low-knowledge learners than for high-knowledge learners and for high-spatial learners rather than for low-spatial learners.

These principles are an appropriate pathway to create learning materials, but should we consider all these principles in designing examples? We have not seen any prior studies using examples that consider all of these principles in one study. It is inevitable that some principles be omitted in favour of having adaptive systems.

One way of scaffolding ITSs is to provide students with worked examples once they submit an incorrect solution. For instance, instead of giving different levels of feedback messages, the system shows an isomorphic worked example of a current problem. This process is relevant to a specific type of example-based activity known as analogical problem solving (APS) (VanLehn, 1998; Muldner & Conati, 2010). Example and problem similarity is categorised via two perspectives: superficial and structural (Novick, 1988). Superficial similarity is identified by comparing features that are not a part of the domain knowledge, such as features that appear in specifications of problems or examples. Students normally choose examples with superficial similarity, without considering structural similarity. Therefore, the examples are not helpful for solving the problem (Novick & Holyoak, 1991; Novick, 1988).

Ringenberg and VanLehn (2006) compared two distinct version of ANDES (Gertner & VanLehn, 2000). In one version they used ITS and in the second they used examples for feedback. They defined two types of self-explanations for examples: derivations answer (e.g. “where did this step come from?”) and procedural explanation (e.g. “why was this step done?”). Students worked with the system as their homework (ten problems) and a post-test was given a few days before the in-class exam. The post-test consisted of problem matching tasks (twenty multiple-choice tasks) for 30 minutes. The experiment had three conditions: a no training condition (students who did not work with the system at home), a hints condition (students received step-focused messages upon solution

submission) and an examples condition (students saw an example as feedback to their help request). The results showed no significant difference between examples and hints conditions in the post-test, while both were superior to the no training condition. However, students in the examples condition attempted fewer problems than students in the hints condition. This suggests that using examples as feedback to the student's solution is more efficient (not effective) than using individualised step-focused feedback messages.

Examples have been used for supporting problem posing. In problem posing, students are encouraged to create problems for themselves. Problem posing is closely related to problem solving and research shows that problem posing significantly influences problem solving (Silver, 1994). However, novices have difficulty in posing problems. Although novice students are provided with examples about composing novel problems, they do not necessarily know the reasons used to generate examples. Recently, Kojima et al. (2013) investigated the advantage of using examples for problem posing. They conducted a study in which students were given examples presenting problem posing. Students were then asked to reproduce problems identical to the given examples. The results show that reproduction appears to be effective for learning from examples in the domain of problem posing.

3.10 Conclusion

We reviewed a large number of articles that focused on using examples in learning from different viewpoints. First, we explained cognitive load theory and how examples work based on this theory. Then we discussed two types of examples: worked examples and modelling examples. Next, we reviewed articles that showed self-explanation effects, specifically those using self-explanation with examples. We then reviewed a number of articles comparing examples with unsupported problem solving and ITSs, followed by an overview on articles that investigated adaptive examples. We briefly examined the prior literature about example design and whether or not using examples is always an ideal choice.

To summarise, using examples in learning is an important teaching technique, especially for novices. Moreover, examples can be scaffolded with meta-cognitive skills

such as SE. If we choose SE wisely after examples, then support them with isomorphic tutored problems, students may obtain a higher learning gain. Moreover, a number of studies have shown the learning advantage of alternating worked examples with problems (e.g. (Sweller & Cooper, 1985; Kalyuga et al., 2001; Mwangi & Sweller, 1998). In line with the cognitive load theory, the learning benefits shown in these studies suggest to use two-step learning processes. That is, it is helpful for students, particularly those who have low prior knowledge, to study an example and maximise their initial learning. Then, learnt material from studying examples can be used to solve isomorphic problems. Two isomorphic tasks have similar structures and/or elements (Sweller & Cooper, 1985; McLaren & Isotani, 2011). In Chapter 5, we discuss a study that compared examples only, problems only and alternating examples and problems while both examples and problems were scaffolded with self-explanation prompts.

Prior research shows that novices learn more from examples than solving problems. An adaptive pedagogical model can help ITSs to provide students appropriate learning tasks (examples or problems). In Chapter 6 we explain an adaptive model for using examples in SQL-Tutor that improves learning more than the original SQL-Tutor.

Chapter 4. Eye tracking

Eye tracking involves determining the point of gaze of a person's eyes on a visual scene (Goldberg & Helfman, 2010). In recent years, eye tracking has been employed in many areas, ranging from usability studies of graphical user interfaces to marketing and psychology. Eye tracking can also be used instead of other input devices. For example, Wang et al. (2006) used eye movement data to allow students to select a topic to study by simply looking at a portion of the screen for a pre-specified time, or answering questions using eye movements. In psychology, researchers have observed different aspects of human cognition, such as intention or plans, using an eye tracker (Rayner, 1995; 1998). An eye tracker is a measurement device most often used for measuring eye movements (Duchowski, 2007). Eye tracking has been largely used in laboratory settings; to make this technology available to private users, ordinary webcams have recently been used to track eye gaze movements (San Agustin et al., 2010).

In order to monitor eye movements, eye trackers normally use one of two eye movement monitoring techniques: position of the eye relative to the head, or the eye's orientation in space (Duchowski, 2007). However, measurements often contain noise, which can be caused by different factors such as participant's head movements, inaccurate calibration, or wearing contact lenses and glasses (Bednarik, 2005; Goldberg & Helfman, 2010). As a result, eye trackers are not always able to collect eye-gaze data while tracking a user's eye (Bednarik, 2005); moreover, eye trackers cannot report samples constantly for a validity check (Kardan & Conati, 2012). Eye trackers report invalid eye gaze data when the user suddenly looks away from the screen. The invalid data can be identified by inspecting a video of the session. Nevertheless, the invalid data should be excluded during the validation of the rest of the data (Kardan & Conati, 2012).

4.1 Eye tracker, Tobii TX300

In this project, we used the Tobii TX300 eye tracker (see Figure 4.1) to capture eye movements of students. Tobii T and X series eye trackers collect data points of eye movements every 16.7, 8.3 or 3.3 ms depending on whether the sampling rate is 60, 120 or 300 Hz. The Tobii machine we used in this research is TX300 (sampling data rate 300

Hz). Participants can move during the tracking session, while the sampling rate maintains at the same accuracy and precision in Tobii TX300 (Tobii Technology, 2010).



Figure 4.1 The Tobii TX300

Tobii allows unobtrusive eye tracking and collects a variety of data that can be analysed using Tobii Studio™ or externally. Before an evaluation session starts, Tobii Studio calibrates the system with the participant's eye movements. In the calibration phase, the eye tracker asks the participant to follow a red marker over a grid of calibration dots. Next, the eye-tracker shows the calibration results, as shown in Figure 4.2. There are eighteen dots (nine dots for each eye). The offset between the calibration of the centre dot and each gaze point is shown by the marks in the middle of the dots. Experimenters can accept the calibration or repeat the calibration phase.

For each data sample, Tobii provides a validity code, specifying where data has been recorded for both eyes (0), one eye only (1 if it is known which eye, or 2 when such information is not available), or whether the gaze data is corrupted (3) or missing (4). It is recommended⁷ to remove all data with the validity score of 2 and above. Tobii Studio

⁷ Tobii studio manual, 2010

(<http://www.tobii.com/en/eye-tracking-research/global/products/software/tobii-studio-analysis->

3.1.4 calculates the quality of recordings by dividing the number of eye-tracking samples that were correctly identified by the number of attempts (the number of times the eye tracker collects eye movement data points). For instance, 100% samples mean that Tobii has found both eyes throughout the recording; 50% means that both eyes were identified only half of the time, or that one eye was identified during the full recording.

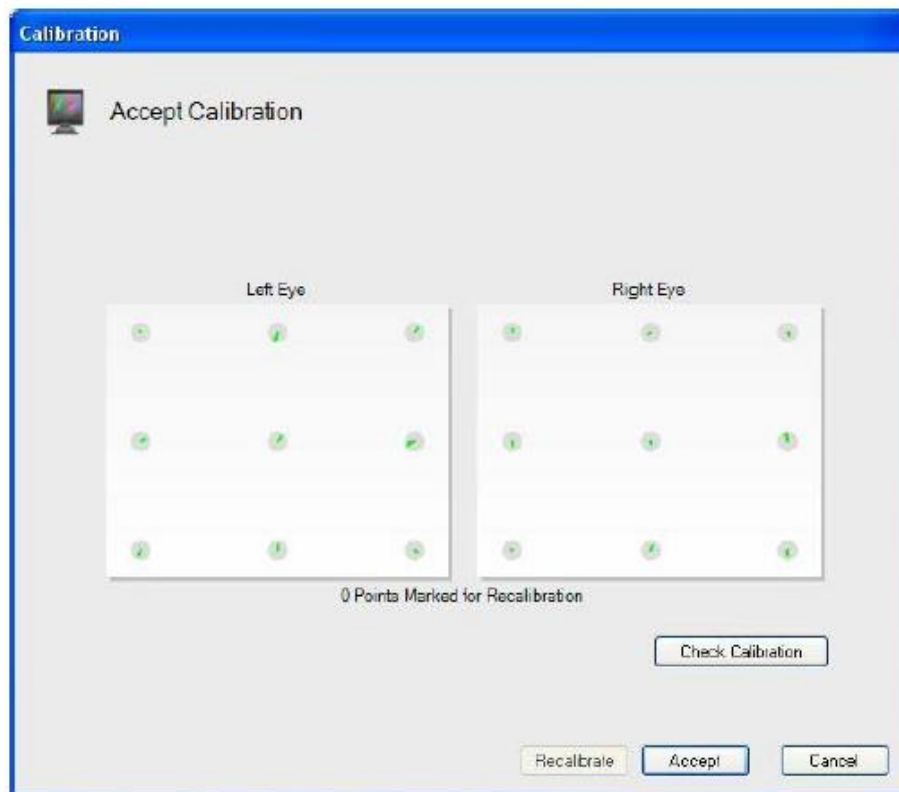


Figure 4.2 A screenshot of the calibration quality screen

The Tobii eye-tracker collects different data, including horizontal and vertical positions of the gaze point for the left and right eyes, validity codes, and size of the pupils, which will be affected by wearing glasses or contact lenses. Afterward, the data can be analysed using Tobii Studio™ or an external tool. For instance, the data can be sent to a web service for analysing in real time (Steichen et al., 2013). Another external tool is the Eye Movement Data Analysis Toolkit (EMDAT), a library developed for processing eye gaze data (Kardan et al., 2012). In this research, we used Tobii Studio™ to analyse eye

gaze data. Tobii Studio™ supports double screens mode, in which an experimenter can monitor the participant’s eye movements on a different screens at the same time.

The human eye makes rapid eye movements (saccades) and fixations on some points on the image (Goldberg & Helfman, 2010). A fixation is defined by selecting a Velocity Threshold value. This value specifies the maximum pixel-distance between two consecutive data points required to be considered a part of fixation. The data points above the threshold are called saccade points and are not used in the analysis. We chose the default threshold in Tobii Studio (defined in the I-VT Filter) because, in contrast with pixel locations of gaze points, the I-VT filter uses the angular velocity of the eyes’ movement data; thus, the data is independent of screen size and resolution, and the distance between the participants and what is shown on the screen (Tobii Technology, 2011). Figure 4.3 shows 13 consecutive fixations captured by Tobii over 3 seconds.

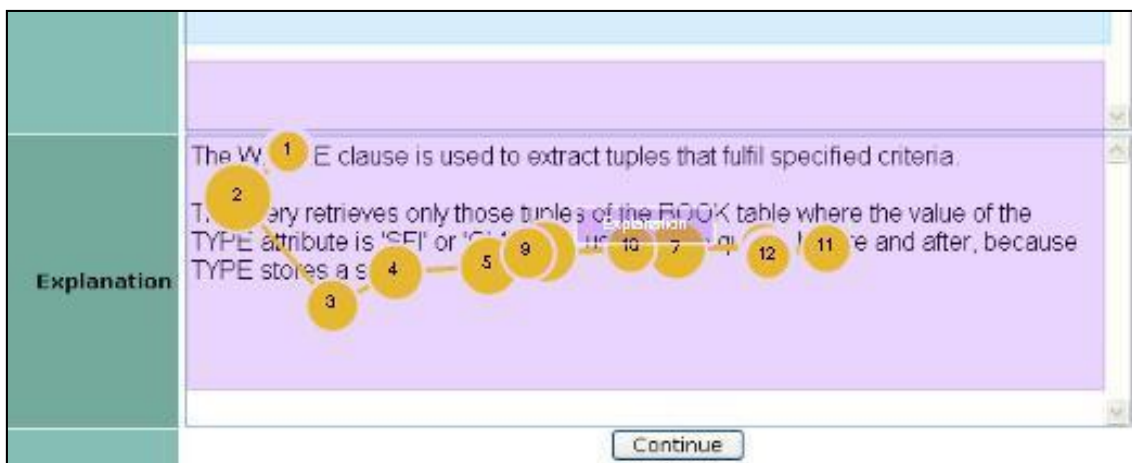


Figure 4.3 A sample screen with fixations captured over three seconds

In Tobii Studio™ we can set up Area of Interest (AOI), which can help with eye-tracking data analysis. In order to distinguish between the concepts of each area’s functionality, we can define AOIs (Kardan & Conati, 2012). There is no restricting rule for setting up AOIs. AOIs help to define scanning sequences and transitions, and also to define which fixations should be tracked (Goldberg & Helfman, 2010). To minimise misclassification of samples of a gaze location, a gap should be used between AOIs where it is possible, otherwise the regions might be too close together (Loboda & Brusilovsky, 2010). For example, in the context of SQL-Tutor interface, we can set up the AOIs as shown in Figure 4.4.



Figure 4.4 An example of AOIs defined for the SQL-Tutor interface

4.2 Eye-tracking literature review

Prior research has employed eye tracking in many areas. In this section, we present an overview of those studies that have used this technology to improve user classification, learning from ITSs and learning from examples.

4.2.1 Eye tracking for user classification (experts or novices)

In these studies, participants were recruited from expert or novice groups. A novice is a participant who has a low prior knowledge about a particular domain, while an expert has a number of years of experience in the domain.

Charness et al. (2001) monitored eye movements of expert and intermediate chess players while they were attempting to select the most appropriate move in five chess positions. Although there was no significant difference in fixation duration, expert players made greater amplitude saccades and fewer fixations per trial than did intermediate players. However, experts made more fixations on empty squares than did intermediate players. Expert players produced a higher proportion of fixations on the relevant pieces than intermediate players when fixating on the pieces. Therefore, the authors conclude that experts perceptually encode piece arrangements instead of individual pieces. As a result, their eye movements are guided by parafoveal or peripheral processing, which help the players to fixate on salient pieces.

Expert and novice pilots also were compared using eye-tracking data during simulated Visual Flight Rules for flight landing (Kasarskis et al., 2001). The study shows that experts produce more fixations than novices; however, experts had significantly shorter fixation duration than novices. The results also show that experts have better defined scan patterns than novices. Because the patterns were consistent and efficient, experts could better maintain airspeed; consequently, they had high landing performance.

Jarodzka et al. (2010) compared the different strategies novices and experts use to describe the locomotion patterns of swimming fish from a video. The result shows that experts attended to the relevant features more than novices and kept their attention on the relevant features. Moreover, experts paid attention to different features as they used knowledge-based patterns unknown to novices.

4.2.2 Eye tracking and ITSs

Research shows that eye tracking can improve user modelling. In Sibert, Gokturk and Lavine's (2000) study, the system tracked reading patterns of students and provided support once the patterns showed difficulties in reading a word. The result showed that the system improved the speed and accuracy of reading. Gluck et al. (2000) demonstrated how eye tracking increased user model bandwidth for educational purposes. Students worked with the modified version of Algebra Tutor (Koedinger & Anderson, 1998). Gluck et al. (2000) found that some students ignore algebraic expressions and/or ignore feedback messages; thus, the tutor can prompt them to pay attention to those areas.

Self-explanation is a meta-cognitive activity in which students explain learning materials to themselves (Chi et al., 1994; Chi et al., 1989). Conati and Merten (2007) found that eye data provides a more accurate assessment of self-explanation compared to using time data only. The result corroborates the findings of Conati et al.'s (2005) study which concluded that eye tracking is superior to time-based evidence for self-explanation, because users who self-explain probably look at AOIs, which indicate the effects of users' exploratory actions.

Amershi and Conati (2007) suggested a user-modelling framework for interactive environments. In the model they used supervised and unsupervised machine learning. First, common learning behaviours were identified for offline initialisation. Users were

clustered using the k-means algorithm (Duda, Hart & Stork, 2012). Clustering was based on feature vectors including eye-tracking data and interface. Then behaviours were labelled as beneficial or detrimental to learning by a human expert. In the online recognition phase, a classifier model used the clusters to categorise students as low achievers or high achievers. Students used the Adaptive Coach for Exploration (ACE), an exploration-based learning environment. ACE allows students to explore mathematical functions (Bunt et al., 2001). The results show that low achievers paused less and made fewer indirect gaze shifts after an action that changes other parts of the interface than high achievers. Later, Amershi and Conati (2009) proposed a data-based user modelling framework. The model used both supervised and unsupervised learning to make student models. They used both logged interface and eye-tracking data. The results showed that the framework could automatically identify meaningful interaction behaviours of students. Moreover, the framework could be used to make student models for the online classification of new student behaviours. Mining of association rules for adaptive actions is a refinement to the framework (Kardan & Conati, 2011).

Kardan and Conati (2012) present two classifiers that identify high and low achievers using students' eye movements. In this study, participants used the Constraint Satisfaction Problems (CFS) applet, an interactive tool for learning Artificial Intelligence algorithms. Participants were asked to solve two problems while an eye tracker was capturing their eye gaze data. The researchers analysed all captured data and also changes in participants' eye movements from the first to the second problem. Overall, their classifier achieved a high accuracy (71%) in classifying students as low or high achievers, using only information about patterns of a user's overall attention. When changes in students' eye movements between the first and the second problem were provided to the classifier, accuracy improved further (76%). In Kardan and Conati's (2012) study, data was captured over the whole session; thus, to achieve a high accuracy, the classifier needed complete session data, otherwise the accuracy of decisions would not be high enough, specifically in the early stages of learning. Kardan and Conati (2013) show that the actions from logs plus eye data improve the classifier's accuracy to 85% by considering 22% of all interaction data such as logs and eye-gaze movements.

Tsai et al. (2012) used eye tracking to examine learners' visual attention while they solved multiple-choice science questions. Students were asked to predict occurrences of landslide hazards from four images illustrating different scenarios. Each scenario showed a combination of four different factors. Researchers investigated the fixation duration between chosen and rejected options and also among the relevant and irrelevant factors. The think-aloud protocol was used in the study; consequently, content analyses were performed to analyse the students' responses and think-aloud protocols. The authors found that students paid more attention to chosen options than rejected options; moreover, they spent more time checking relevant factors than irrelevant factors. Successful participants paid more attention to relevant factors while unsuccessful students experienced difficulty in recognising relevant factors.

Elmadani et al. (2013) investigate how students interact with tutorial dialogs in EER-Tutor by analysing interaction logs and eye-gaze data. EER-Tutor is a constraint-based tutor which teaches conceptual database design (Zakharov et al., 2005). The authors found that advanced students selected appropriate areas to visually focus on, whereas novices paid attention to irrelevant areas on the screen.

4.2.3 Eye tracking and learning from examples

Several studies have employed eye tracking to improve learning from examples, mostly focusing on how to guide the user's attention via experts' eye movements. Van Gog et al. (2009) showed that when the eye movement guidance was combined with a verbal explanation, it had a detrimental effect on learning. This is surprising since, based on the modality principle in multimedia learning, learners learn more when they use both visual and audio channels for learning (Mayer, 2002).

Jarodzka et al. (2013) investigated a new technique to teach perceptual tasks. They used eye movement modelling examples (EMME) to teach classifying fish locomotion. In modelling examples, the problem solution was demonstrated to learners by a model, which could be an animated agent or a human (van Gog & Rummel, 2010). They had three conditions: EMME with dot display, EMME with spotlight display and a control group. Students in EMME conditions received attention guidance by using dots or a spotlight on the screen to show where the expert was looking. Students who saw the

spotlight could not see irrelevant areas as they were opaque on the screen. The control group did not receive attention guidance. Students first watched four videos in which the expert explained how to classify fish locomotion, while students in the EMME conditions could additionally see where the expert was looking on the screen. Next, the students watched four videos for a visual search test and finally took an interpretation test. Overall, the study showed that EMME improves visual searching and enhances interpretation of relevant information. A few contrasts between EMME conditions show that the dot and spotlight displays have different effects on various test features. For instance, the spotlight group had a first fixation earlier than the dot display condition. On the other hand, the dot display group outperformed the spotlight group in an interpretation question which measured the ability of the group in memorising the essential fish parts.

Litchfield et al. (2010) conducted three experiments to observe how guiding attention via other people's eye movements would improve radiographer performance in reading chest X-rays. In Experiment 1 they found no significant difference between the performance of the groups given eye movements of a novice or an expert radiographer. In Experiment 2, only novices improved when they were provided with the expert's eye movements. In Experiment 3 they restudied the contribution of image, task and level of models' expertise. Litchfield et al. (2010) suggest that guiding users' attention via another person's eye movements may have a short-lived effect, but it can help novices to scaffold their decision using other people's search behaviour.

4.3 Conclusion

We explained eye tracking and reviewed articles that have used eye tracker to improve learning from ITSs and learning from examples.

We believe that novices and advanced students study examples differently. Therefore, not only the content of examples should be adapted to the learner's need, but also the order of the examples' content should encourage novices to study examples in a way that advanced students learn. Moreover, ITSs can provide feedback messages to draw student's attention to the relevant areas of examples. In order to get deeper insights about how students study examples, we conducted a study using eye tracking. The study is presented in Chapter 7.

Chapter 5. First study: Worked-example effect in SQL-Tutor

The results of the study described in this Chapter were published in (Shareghi Najar & Mitrovic, 2013a; Shareghi Najar & Mitrovic, 2013b). The study aimed to reduce cognitive load by using scaffolding. We found that using a mixture of examples and tutored problem solving was superior to problem-solving only and examples only.

Previous studies have shown that learning from worked examples is superior to unsupported problem solving. Examples reduce the cognitive load on the learner's working memory, thus helping the student to learn faster or deal with more complex problems. Sweller and Cooper (1985) explained a two-step learning process. First, examples are a suitable approach for students, particularly for novices, since examples reduce the cognitive load and increase the initial learning. Second, students use the knowledge they learnt from studying examples in solving similar problems. Therefore, it is logical that novices benefit the most from studying examples, but when novices' knowledge increases, examples eventually must be transferred to problems. That is, students with different prior knowledge levels benefit differently from the two learning tasks (examples and problems).

ITSs are different from unsupported problem solving as they provide adaptive scaffolding in terms of feedback, guidance, problem selection and other types of help. Only recently several studies have compared learning from examples to learning with ITSs (Schwonke et al., 2009; McLaren & Isotani, 2011; Kim et al., 2007; Anthony et al., 2008; McLaren et al., 2008; Salden et al., 2010). However, a number of studies that compared examples with ITSs show that studying from examples only is superior to working with an ITS. ITSs can be transferred to examples upon a complete solution request by students (McLaren et al., 2008); moreover, large amounts of information offered by examples may cause expertise reversal for advanced students (Kalyuga, 2007).

We thought that for the following reasons we had to investigate the worked-example effect in SQL-Tutor:

- All the prior studies that compared examples with ITSs used domains with well-defined tasks. Therefore, we needed to compare learning from examples and problems in the SQL domain which has ill-defined tasks.
- None of the prior research compared worked examples with a constraint-based tutor. SQL-Tutor is a constraint-based tutor.
- In one of the prominent studies (McLaren & Isotani, 2011) that compared examples with tutored problem solving, researchers used Self-Explanation (SE) prompts after examples. It has been evident that SE improves learning. Therefore, we believe that problems also had to be scaffolded with SE.
- A number of studies showed that using examples is a more efficient choice than ITSs to improve learning gain. The results, however, are not conclusive, as the domain and study design influence outcomes of studies.

Bearing in mind the above reasons, we conducted a study that compared learning from examples only (EO), alternating examples and tutored problems (AEP), and tutored problems only (PO) in SQL-Tutor. We scaffolded examples and problems with SE prompts which required students to explain the principles necessary for solving a problem or explain how the solution was generated. Because of the two-step learning process, our hypothesis was that the AEP condition would be superior to both PO and EO conditions. Students in the AEP condition have a chance to learn the material from examples and rehearse what they have learnt in solving problems. In addition, we hypothesised that novices and advanced students would learn less from EO than from AEP/PO.

We start by describing the material, participants and the procedure of the study, while Section 5.2 presents the results of the study. The discussion and conclusion are presented in Section 5.3.

5.1 Experiment design

In our project, we focus on defining database queries using the Structured Query Language (SQL). This instructional task is more complex than learning tasks used in prior research, and is also ill-defined tasks. We performed the experiment with web-based SQL-Tutor (Mitrovic, 2003), to which we added examples and self-explanation features (SQL-Tutor was described in Section 2.4.) We disabled the Open Student Model (OSM)

and problem selection functions because we did not want other learning factors to affect our study. Figure 5.1 shows the architecture of the version of SQL-Tutor used in this study.

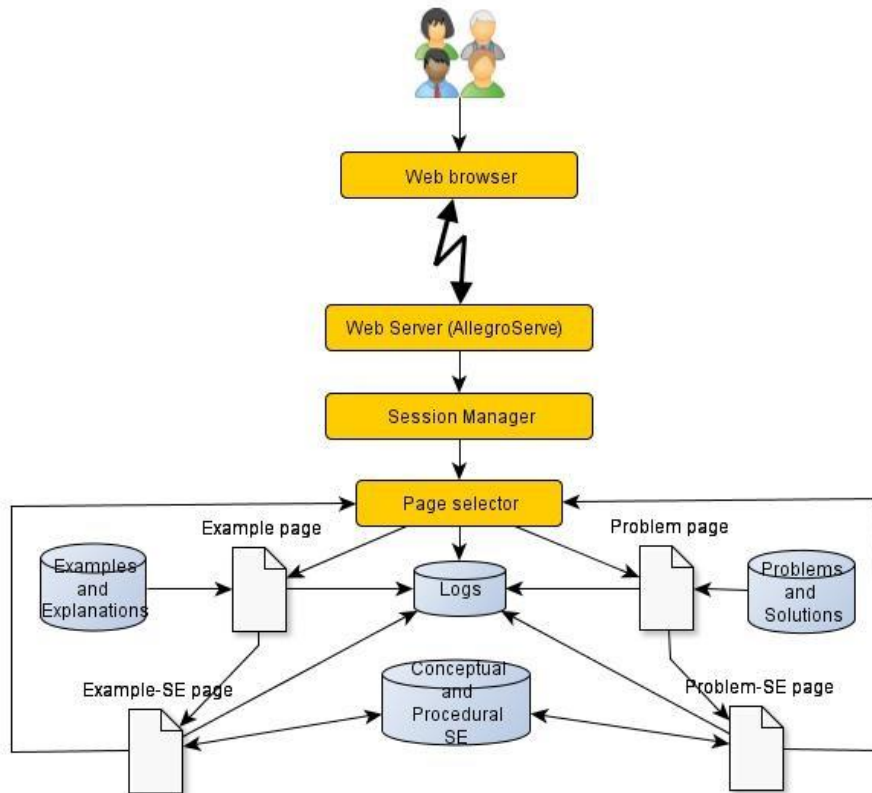


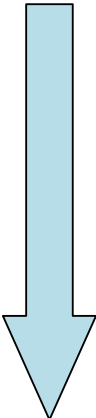
Figure 5.1 The architecture of the system

We chose the *CD collection* database from thirteen databases available in SQL-Tutor. Figure 5.2 shows the database schema. Primary keys in the attribute list are underlined and foreign keys are in italics. Students had access to the database schema at any time during the examples studying, problem solving and self-explaining.

ARTIST	(<u>ID</u> , LNAME, FNAME)
IN_GROUP	(<u>GROUP_NAME</u> , <i>ARTIST</i>)
CD	(<u>CAT_NO</u> , TITLE, YEAR, PUBLISHER, <i>GROUP_NAME</i> , <i>ARTIST</i>)
SONG	(<u>ID</u> , TITLE)
COMPOSER	(<u>ID</u> , LNAME, FNAME)
SONG_BY	(<i>SONG</i> , <i>COMPOSER</i>)
RECORDING	(<u>ID</u> , SONG, DATE, LENGTH)
CONTAINS	(<i>CD</i> , <i>REC</i>)
PERFORMS	(<i>REC</i> , <i>ARTIST</i> , INSTRUMENT)

Figure 5.2 The schema of the CD collection database

For this study, we developed three versions of SQL-Tutor which provided different combinations of worked examples and problems. Figure 5.3 shows the study design. In each condition, the student was given a set of 20 problems/examples arranged in pairs (given in Appendix A.1), 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. Each student was randomly assigned to a group (EO, AEP or PO). The page selector component decided on the next page (example or problem) based on the student's group.



PO	AEP	EO
Pre-test		
20 problems in 10 isomorphic pairs	20 problems and examples in 10 isomorphic pairs	20 examples in 10 isomorphic pairs
1 st in each pair: problem 2 nd in each pair: problem	1 st in each pair: example 2 nd in each pair: problem	1 st in each pair: example 2 nd 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 example followed by a P-SE prompt
Post-test		

Figure 5.3 Design of the study with three conditions

Previous research showed that worked examples increase conceptual knowledge more than procedural knowledge, while problem solving produces results in higher acquisition of procedural knowledge (Schwonke et al., 2009; Kim et al., 2007). 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 5.4 shows a sample of a problem/example with correlated explanation, C-SE and P-SE prompts.

Problem/Example:

Show the names of all groups in descending order.

```
SELECT DISTINCT group_name
FROM in_group
ORDER BY group_name DESC
```

Explanation:

Some attributes in a table may contain duplicate values. However, sometimes you may want to list only different (distinct) values from a table. The DISTINCT keyword can be used to return only distinct values.

The ORDER BY clause is used to sort the result-set by a specified attribute. The ORDER BY clause sorts the records in ascending order by default (or using ASC). Use the DESC keyword when you want to sort the records in a descending order.

Procedural self-explanation:

What will happen if we don't use DISTINCT in this example?

- A In that case all attributes will be selected.
- B Only unique tuples will be selected.
- C Then the number of tuples may become larger than the number of groups.
- D The system gives an error.

Feedback

- A Wrong - all attributes will be selected if we use * in the SELECT clause. If we do not use DISTINCT, all values (including duplicates) will be retrieved.
- B Hmm, that's not the answer. Actually only with using DISTINCT duplicates will not be selected. If we do not use DISTINCT, all values will be retrieved.
- C Yes, that's the answer. Without DISTINCT, the query may return more tuples, and some group names may be shown more than once in the query output.
- D No - although the result will be wrong, the system doesn't give an error. If we do not use DISTINCT, all values (including duplicates) will be retrieved.

Conceptual self-explanation:

What do the DESC and ASC keywords do in an ORDER BY clause?

- A ASC avoids selecting duplicates.
- B ASC sorts the records in a descending order, and DESC in ascending order.
- C DESC avoids selecting duplicated records.
- D DESC sorts the records in a descending order, and ASC in ascending order.

Feedback

- A That is incorrect – ASC is used to sort the resulting tuples in the ascending order.
- B No, it's the opposite way.
- C Your answer is incorrect - DESC is used to sort the resulting tuples in the descending order.
- D Good job!

Figure 5.4 Sample of a problem/example, and corresponding C-SE and P-SE prompts

Error! Reference source not found. shows a comparison between two new types of prompts in this study and different types of SE prompts used in prior research. Most of prior studies used audio recordings to evaluate self-explanations. However, we followed the approach McLaren, Isotani (2011) used in their study as that study had highest similarity with our study. McLaren and Isotani (2011) used one type of explanation, and students only received self-explanation prompts after examples. Table 5.1 also shows that similar to our study, Alevan, Koedinger and Cross (1999) provided SE after problems. Based on definitions we provided for P-SE and C-SE, Alevan and colleagues used P-SE after problems. As we discussed in Section 3.3, Chi et al. (1994), Chi and VanLehn (1991), and Haussmann et al. (2009) suggested eight new types of self-explanation. Those eight SE types and our suggested SE types have some overlaps, but the only definitions that emphasise on conceptual knowledge and procedural knowledge are C-SE and P-SE.

Figure 5.5 contains a screenshot of a worked-example page. The complete example was shown at the same time. The problem and the solution are shown at top left of the screen and the explanation about the example is shown below the solution. The information about the interface is shown on the right side of the screen. At the top of the screen, students could click on ‘History’ and ‘Log out’ buttons. The History button allowed students to review examples they had studied or problems they had solved. Therefore, students had access to all the previous solutions they submitted and the SE prompts they had already answered during the session. Students had to click on the Log out button to finish the learning session. Students could click on the continue button below the explanation to proceed to the next page, which answered a P-SE prompt.

Figure 5.6 shows a screenshot of a situation when a student finished reading an example. Next, the system showed a P-SE prompt, located on the right side of the screen. The student gave a correct answer to the prompt and the system provided positive feedback. Students had only one attempt for each P-SE prompt. If the student’s P-SE answer was incorrect, the system disclosed the correct answer and let the student to continue using the ‘Next’ link.

Table 5.1 C-SE and P-SE compared to self-explanation used in in prior research

	Comments	Type of SE	Procedural focused or conceptual focused	Domain
Our study	Multiple choice C-SE after problem P-SE after examples	C-SE (by ITS): What do the DESC and ASC keywords do in an ORDER BY clause? P-SE (by ITS): What will happen if we don't use DISTINCT in this example?	C-SE is conceptual focused P-SE is procedural focused	SQL
Chi et al. 1994	Audio recording After reading a declarative texts	General prompt (by examiner): What does this sentence mean? Function prompt (on the text): What does this part do Clarification prompt (by examiner); when student answer was vague: Can you explain that?	General prompt: the explanation can be procedural or conceptual focused. Function prompts are conceptual focused. Clarification similar to general prompt is neither conceptual nor procedural focused.	circulatory system
McLaren, Isotani, 2011	Multiple-choice pull, down menu After examples	3 to 5 SE prompts: "We used the unit conversion term 1 kL solution / 1000 L to convert."	From samples provided in the paper: Procedural-focused SE	Stoichiometry
Chi and VanLehn (1991)	Audio recording	Procedural explanation (e.g. answer to "why was this step done?") Derivation SE (e.g. answer to "where did this step come from?")	Based on our definition, Derivation is under procedural-focused SE	Physics
Hausmann et al. (2009)	Think-aloud protocol After examples	Justification-based prompts (e.g. "what principle is being applied in this step?") Meta-cognitive prompts (e.g. "what new information does each step provide for you?") Step-focused prompts (e.g. "what does this step mean to you?")	All three types are conceptual-focused SE Meta-cognitive and self-focused prompts can focus on procedures too.	Physics
Aleven, Koedinger, & Cross, 1999	Selecting relevant rule After problems	Some rules dealing with isosceles triangles are highlighted in Glossary. Which of these reasons is appropriate?	Procedural-focused	Geometry

A typical problem-solving page is shown in Figure 5.7. The problem is isomorphic to the example shown in Figure 5.5. The problem-solving environment is similar to the original SQL-Tutor: students provide answers after clauses then submit their solutions. The system shows feedback messages on the right side of the screen. Students had to answer problems correctly to go to the next step. Once students submitted a correct solution, the system showed a C-SE prompt on the right side of the screen.

Figure 5.8 shows a screenshot of a problem-solving task, but 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 negative feedback and revealed the correct answer. Similar to P-SE prompts, students had only one attempt for each C-SE prompt. Once students received SE feedback, they could continue with the next task.

As you can see in Figures 5.5, 5.6, 5.7 and 5.8, the version of SQL-Tutor used in the study excluded interface features that were not required (for example, the button to bring up the open student model).

Participants were 34 students enrolled in the Relational Database Systems course at the University of Canterbury. The study was conducted during the eighth week of the semester. Students had learnt about SQL in lectures beforehand and needed to practise in the lab. The students did not receive any inducements for participating in the study, but we told them that working with our system will help them learn SQL. We informed them that they would see ten pairs of problems and that the tasks in each pair were similar. We believe if students know that the tasks in each pair are isomorphic, they may use them more efficiently.

The study was conducted in a single, 90-minute session. At the beginning of the session, the students took a pre-test (given in Appendix B.1) for 10 minutes. Once 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 a post-test for 10 minutes at the end of the session.

SQL-TUTOR		History	Log Out
Example 7	<p>Show the titles of songs composed by George Gershwin.</p> <pre>SELECT title FROM composer, song_by, song WHERE song = song.id and composer.id =composer and lname = 'Gershwin' and fname = 'George';</pre>	<p>The buttons in the navigation bar at the top of page have the following functionality:</p> <p>"History" button shows a brief history of the current session.</p> <p>Click the "Log Out" button when you want to finish the session.</p> <p>The bottom section of the page shows the database schema. You can see how many tables there are, and what their names and attributes are. More information about the database is available by following the links shown there.</p> <p>An example problem with its solution and explanation is provided for you. Please read them carefully and think about the solution. Click the "Continue" button when you are finished.</p>	
Explanation	<p>The WHERE clause can contain many conditions, which are used to retrieve only some of the tuples from the given tables.</p> <p>If two attributes from two tables have the same name, then we have to use qualified names (table_name.attribute_name).</p>		
<input type="button" value="Continue"/>			

Figure 5.5 Screenshot of a worked-example page

SQL-TUTOR History Log Out	
Example 7	<p>Show the titles of songs composed by George Gershwin.</p> <pre> SELECT title FROM composer, song_by, song WHERE song = song.id and composer.id =composer and lname = 'Gershwin' and fname = 'George'; </pre>
Explanation	<p>The WHERE clause can contain many conditions, which are used to retrieve only some of the tuples from the given tables. If two attributes from two tables have the same name, then we have to use qualified names (table_name.attribute_name).</p>
<p>In the WHERE clause of the given example, which criteria join the three tables?</p> <p> <input type="radio"/> A) lname='Gershwin' and fname='George' <input type="radio"/> B) fname='George' <input type="radio"/> C) lname='Gershwin' <input checked="" type="radio"/> D) song=song.id and composer.id=composer </p> <p style="color: green;">Well done!</p>	
Next	

Figure 5.6: Screenshot of an example page followed by P-SE

SQL-TUTOR		History	Log Out
Problem 8	Show the surnames of artists in the 'Queen' group, as well as the titles of their CDs.		<p>Almost there - you made 3 mistakes.</p> <p>Semicolon (;) specifies the end of the query, and therefore cannot appear in the SELECT clause.</p> <p>You can correct your query and press 'Submit' again, or try getting some more feedback.</p> <p>Would you like to have another go?</p>
SELECT	lname;title		
FROM	artist, in_group, cd		
WHERE	artist.id= in_group.artist and in_group.group_name='Queen' and CD.group_name= in_group.group_name		
GROUP BY			
HAVING			
ORDER BY			
Feedback Level	Hint	Submit Answer	

Figure 5.7 Screenshot of a problem-solving page

SQL-TUTOR		History	Log Out
Problem 8	Show the surnames of artists in the 'Queen' group, as well as the titles of their CDs.		
SELECT	lname, title		
FROM	artist, in_group, cd		
WHERE	artist.id= in_group.artist and in_group.group_name='Queen' and CD.group_name= in_group.group_name		
GROUP BY			
HAVING			
ORDER BY			

When do we need to use qualified names for attributes in the WHERE clause?

A) a sorted result is needed

B) attributes from two different tables have the same name

C) tables are not specified in the FROM clause

D) the result should be grouped.

No - tables are always specified in the FROM clause. Please see the correct answer.

[Next](#)

Figure 5.8 Screenshot of a problem-solving page followed by C-SE

The pre-test had five questions, three of which were multiple-choice (one point each) and two were problem-solving questions (four points each). The first two multiple-choice questions measured the conceptual knowledge students had, while the third question measured their procedural knowledge. For the fourth and the fifth questions, students had to write SQL queries. These two questions measured procedural knowledge and the problem-solving skill of the students. The post-test (given in Appendix B.2) 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.

5.2 Results

We used the software package SPSS 19 to compare the groups and Microsoft Excel for the basic analysis and diagrams. We calculated the average scores in the pre- and post-tests, and the time students spent on the system (Table 5.2). Students who had pre-test scores lower than 45% were considered novices and the rest were classified as advanced students.

Table 5.2 Basic statistics for all students (standard deviation given in brackets)

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

We analysed the data to find the answers to two questions: How did students learn from the three conditions? How did novices and advanced students benefit from different versions of the system? We start by explaining the results for all students followed by explaining the results for novices and advanced students.

5.2.1 Results for all students

The basic statistics about the study are presented in Table 5.3. There was no significant difference between the pre-test performances of the three groups. ANOVA revealed a significant difference between the post-test results ($p = 0.02$). The Tukey post-hoc test

showed that the performance of the EO group was significantly lower than the AEP group ($p = 0.02$) and marginally significantly lower than the PO group ($p = 0.09$), thus confirming our hypothesis. The students in all three conditions improved significantly between the pre- and post-test, as shown by the paired t-test reported in the Improvement row of Table 5.3. Correlations between the pre- and post-test scores are also reported in Table 5.3, but only the PO condition had a significant correlation ($p = 0.01$, $r = 0.69$).

Table 5.3 Basic statistics for the three conditions (* 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)	0.48
Post-test (%)	72.73 (13.98)	77.69 (16.57)	58.68 (16.57)	*0.02
Improvement	*p=0, t=-9.8	*p=0, t=-5.1	*p=0.03, t=-2.4	
Pre/post-test correlation	*p=0.01, r=0.69	p=0.49, r=0.22	p=0.43, r=0.26	
Learning time (min)	69.67 (11.16)	65.91 (14.53)	38.45 (16.14)	*<0.01
Number of attempted problems	14.58 (5.11)	14.09 (5.10)	18.63 (3.23)	0.05
learning gain^N	0.54 (0.19)	0.55 (0.31)	0.21 (0.35)	*0.01
Problem solving gain^N	0.64 (0.27)	0.58 (0.42)	0.19 (0.37)	*0.01
Conceptual knowledge gain^N	0.29 (0.39)	0.77 (0.41)	0.54 (0.47)	*0.03
Procedural knowledge gain^N	0.59 (0.22)	0.48 (0.42)	0.13 (0.40)	*0.01
Perceived task difficulty	3.50 (0.80)	3.27 (0.90)	2.82 (0.75)	0.15

There was also a significant difference between the mean learning times of the three groups ($p < 0.01$). 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 $p < 0.01$). 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 ($p = 0.05$). The Tukey post-hoc test revealed that the EO group attempted more tasks than PO ($p = 0.1$) and the AEP group ($p = 0.07$).

The three groups also differed significantly in the normalised learning gain⁸ ($p = 0.01$). The Tukey post-hoc test revealed that the EO group learned significantly less than students in the AEP group ($p = 0.02$) and the PO group ($p = 0.03$). When we analysed normalised learning gains on the problem-solving questions only (questions 4 and 5), we found a significant difference between the groups ($p = 0.01$). 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 $p = 0.01$, EO and AEP $p = 0.04$), because students in the EO condition were not given any problem-solving tasks during the learning phase.

We also analysed how students acquired 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 in both conceptual and procedural normalised learning gain. The Tukey post-hoc test revealed that the AEP group learned significantly more conceptual knowledge than the PO group ($p = 0.02$). 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 ($p = 0.01$); the Tukey post-hoc test revealed a significant difference between the PO and EO conditions ($p = 0.01$) and a marginally significant difference ($p = 0.06$) between the AEP and EO conditions.

In the post-test we also asked students about the perceived task difficulty. The Kruskal-Wallis Test show no significant difference between the three conditions ($p = 0.15$). 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 was marginally significant ($p = 0.053$). This result was expected as problems impose more cognitive load on the working memory than examples (Sweller et al., 2011).

⁸ Normalised learning gain = (Post test - Pre test) / (Max score - Pre test); in the tables, (^N) represents normalised gain of a variable

We calculated the effect size based on the normalised learning gain using Cohen's d , with following assumption: $d \geq 0.8$ (large effect), $d \geq 0.5$ (medium effect) and $d \geq 0.2$ (small effect) (Cohen, 1988). The result is reported in Table 5.4. The effect sizes for both the AEP and PO conditions were large in comparison to the EO condition.

Table 5.4 The effect size on normalised learning gain between the groups

Conditions		Effect size
AEP	PO	0.04
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 EO participants were given. We also analysed the SE success rates for the three conditions, which are reported in Table 5.5. 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 5.5 SE prompts analysis

	PO	AEP	EO	p
C-SE success rate (%)	88.50 (7.5)	92.84 (10.36)	N/A	0.26
P-SE success rate (%)	N/A	77.69 (19.74)	71.36 (11.20)	0.37

Students in the PO and AEP groups could select the feedback level 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 5.6 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 5.6) 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 ($p =$

0.39). This result shows the participants from the PO/AEP groups did not convert their problems to worked examples.

Table 5.6 Maximum hint level analysis

	PO	AEP	
Second problem in pairs	1.08 (1.68)	1.54 (1.69)	p = 0.51
First problem in pairs	1.33 (1.56)		
	p = 0.39		

5.2.2 Learning curves

Plotting learning curves is a method to investigate how the students in AEP and PO learnt SQL concepts in terms of constraints. For this, the number of times the constraints were relevant was plotted against the occasions when they were used incorrectly. A good fit to the power curve should result if the constraint measured is being learnt (Martin & Mitrovic, 2005).

For this analysis, we excluded the EO group as they worked with examples only; consequently, they did not violate any constraints. Constraints are violated when a student submits an incorrect solution to a problem. Figure 5.9 shows the learning curves for the PO and AEP conditions. The PO graph has a good fit to the power curve ($R^2 = 0.80$), but the fit to the AEP graph is poor ($R^2 = 0.56$). As shown by the slope, the learning rate is higher for the PO trend line. Note that students in AEP saw an isomorphic example before solving a problem; therefore, they violated fewer constraints (as they had learnt from the example).

We also investigated the number of constraints learned by students in AEP and PO. For each constraint in the student model, we considered the first five attempts and the last five attempts when the constraint was relevant. The constraint was considered to be learnt if the probability of violating a constraint was reduced by 70% during the last five attempts (Weerasinghe et al., 2010). We used the t-test to compare the number of constraints students learnt in AEP and PO. Table 5.7 shows the result. We found no significant difference between AEP and PO in the number of constraints students learnt. A possible reason is that AEP might learn constraints from examples; thus, they did not violate the constraints when they solved the subsequent problems.

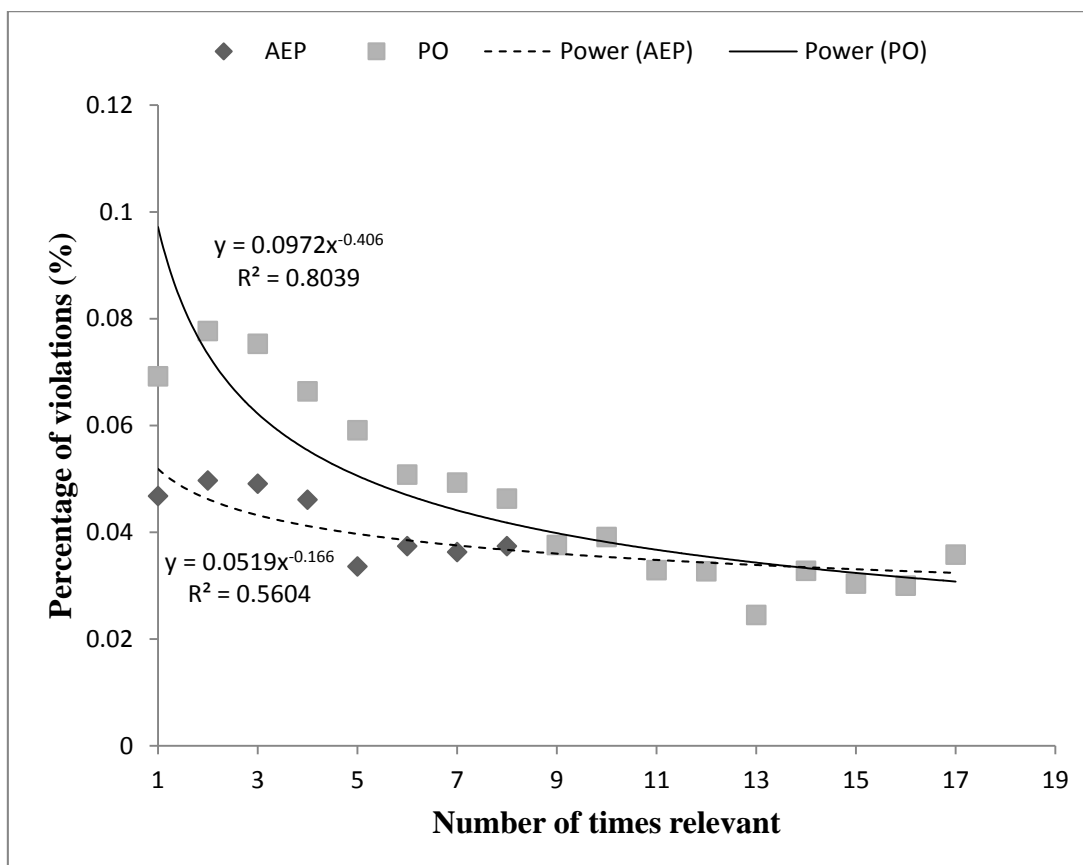


Figure 5.9 The probability of constraint violations for PO and AEP

Table 5.7 The number of constraints learnt

	PO	AEP	
Constraints learnt	7.75 (2.45)	3.09 (3.78)	p = 0.17

5.2.3 Results for novices and advanced students

In this section, we present results for novices and advanced students in PO, AEP and EO. We compared novices then advanced students in the three conditions.

Table 5.8 presents some statistics about the novices. The Kruskal-Wallis 1-way ANOVA test did not reveal a significant difference on the pre-test performance of novices from the three conditions; therefore, our groups were comparable. Using the same test, we found no significant difference between the groups on the post-test. The Wilcoxon signed-rank tests show that novices in the PO and AEP conditions improved significantly between the pre- and post-tests, while EO showed a marginally significant improvement. There were no significant differences between the three conditions on the normalised

learning gain, but there was a marginally significant difference in learning time ($p = 0.06$). The Mann-Whitney test showed that novices in EO spent significantly less time than novices in AEP ($p = 0.03$) and PO ($p = 0.05$). The table also indicates a significant difference in the normalised conceptual knowledge gain ($p = 0.04$) and the Mann-Whitney test revealed that novices learnt significantly more conceptual knowledge from AEP than PO ($p = 0.01$).

Table 5.8 Dependent variables for novices (^N Normalised)

	PO	AEP	EO	p
Number of students	6	5	5	
Pre-test score (%)	31 (11)	36 (5)	33 (10)	0.79
Post-test score (%)	65 (12)	73 (18)	53 (14)	0.14
Improvement pre- to post-test	$p = 0.03^*$	$p = 0.04^*$	$p = 0.07$	
Learning gain ^N	0.50 (0.14)	0.56 (0.30)	0.29 (0.20)	0.13
Learning time (min)	67 (12)	70 (12)	46 (15)	0.06
Multiple choice questions ^N	0.33 (0.27)	0.60 (0.09)	0.20 (0.68)	0.15
Problem solving questions ^N	0.56 (0.22)	0.55 (0.44)	0.24 (0.35)	0.28
Conceptual knowledge ^N	0.42 (0.38)	1.00 (0)	0.70 (0.45)	0.04*
Procedural knowledge ^N	0.52 (0.20)	0.45 (0.36)	0.18 (0.29)	0.12

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 5.9. The Kruskal-Wallis 1-way ANOVA test shows a significant difference between novices in the three conditions on the overall success rate. The Mann-Whitney test reveals that novices in PO and AEP scored significantly higher than novices in EO ($p < 0.01$ and $p = 0.03$). Moreover, the Mann-Whitney test indicates a significant difference in P-SE success rate on SE prompts ($p = 0.3$); 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 EO. We think that the ITS engaged novices with both examples and problems, while examples could not provide any rehearsal opportunity. On the other hand, AEP novices learned 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 5.9 Analysis of SE performance for novices

	PO	AEP	EO	p
SE success rate (%)	88 (7)	87 (12)	67 (8)	0.02*
C-SE success rate (%)	88 (7)	90 (14)	N/A	0.26
P-SE success rate (%)	N/A	85 (12)	67 (8)	0.03*

Students who scored more than average in the pre-test were classified as advanced students and their performance is reported in Table 5.10. 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-rank tests revealed that advanced students in EO did not significantly improve between the pre-test and the post-test ($p = 0.42$). The table shows a marginally significant difference in 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 ($p = 0.04$) and learning time ($p < 0.01$). This result is in line with those studies that show advanced students learn more from problem solving only than from reviewing examples only. The Mann-Whitney test also shows a significant difference between EO and AEP on learning time ($p = 0.02$). 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 may be caused by an illusion of understanding.

We analysed the performance of advanced students on SE prompts, summarised in Table 5.11. A 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 5.10 Dependent variables for advanced students (^N Normalised)

	PO	AEP	EO	p
Number of Students	6	6	6	
Pre-test (%)	52 (6)	59 (7)	55 (10)	0.16
Post-test (%)	80 (13)	82 (15)	64 (18)	0.16
Improvement pre- to post-test	p = 0.03*	p = 0.03*	p = 0.42	
Learning gain	0.59 (0.24)	0.55 (0.36)	0.15 (0.46)	0.23
Learning time (min)	73 (10)	63 (17)	32 (15)	<0.01*
Multiple choice questions ^N	0.17 (0.26)	0.50 (0.44)	-0.03 (0.82)	0.34
Problem solving questions ^N	0.72 (0.32)	0.61 (0.45)	0.16 (0.42)	0.08
Conceptual knowledge ^N	0.17 (0.40)	0.58 (0.49)	0.42 (0.49)	0.28
Procedural knowledge ^N	0.66 (0.26)	0.52 (0.50)	0.08 (0.50)	0.12

Table 5.11 Analysis of SE prompts for advanced students

	PO	AEP	EO	p
SE success rate (%)	89 (8)	83 (12)	75 (13)	0.18
C-SE success rate (%)	89 (8)	95 (6)	N/A	0.22
P-SE success rate (%)	N/A	72 (24)	75 (13)	0.94

Overall, we found that novices improved the most from the AEP condition in comparison to the other two conditions. Moreover, advanced students did not improve when learning from examples only; therefore, EO was not an appropriate approach for them.

As we mentioned before, students could transform a problem-solving task to a worked example by asking for the complete solution. Therefore, we analysed the help requests submitted for the problems given to the PO and AEP conditions. Similar to the result for all students (Table 5.6), there was no significant difference between novices and advanced students (in PO and AEP), in numbers of requests for complete solutions.

5.3 Discussion and conclusion

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 to ITSs indicate that students learn the same from worked examples and ITSs in domains with well-defined tasks. This encouraged us to observe the worked-example

effect in a domain with ill-defined tasks (SQL). We compared students' performance in three conditions: alternating example/problems, problems only and examples only. We analysed the data to find how students in general benefit from different versions of the system and how novices and advanced students improve on each condition.

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 for all students 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.

The results show that novices who worked with alternating examples and problems, or problems only out-performed novices who worked with examples only. This suggests that novices benefit most when they are engaged in tutored problem solving. On the other hand, the results showed 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 using what they have learnt 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, and only require practice in applying knowledge by solving new problems. The EO condition did not have problem-solving opportunities. Moreover, examples might cause the expertise reversal effect for advanced students (Kalyuga et al., 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., 2009b).

The results show that students who worked with examples did not learn the same as students who worked with problems only and alternating examples/problems. Our result

is in contrast with the findings presented in (McLaren & Isotani, 2011). There are three main differences between the two studies. First, in our study the participants were given self-explanation prompts after examples and problems, not only after worked examples as in (McLaren & Isotani, 2011). Moreover, we designed self-explanation prompts to complement problem solving and examples. We provided procedural-focused self-explanation prompts after examples, as examples have been shown to reinforce conceptual knowledge more than procedural knowledge. We also provided conceptual-focused self-explanation prompts after problem solving to reinforce the acquisition of conceptual knowledge. Therefore, both types of self-explanation prompts were designed 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 specifying SQL queries. Thirdly, our constraint-based tutor provided feedback on demand, while the Stoichiometry tutor used in (McLaren & Isotani, 2011) 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 avoids impasses and is thus less frustrating and more effective. Examples may also induce an illusion of understanding. 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. For example, Roll et al. (2011) describe the Self-Assessment Tutor, an ITS to improve the accuracy of students' judgments regarding their own knowledge.

As all of the prior research compared examples with ITSs in domains with well-defined tasks, we investigated the worked-example effect in SQL, a domain with ill-defined tasks. Worked examples for ill-defined tasks do not disclose all possible answers to problems because there is no a certain procedure available to find solutions. For

instance, an example of an essay is not the only good essay. P-SE that we used in this study encouraged students to explain a step in a solution. As tasks were ill-defined, the nature of our P-SE prompts was different from SE prompts used in previous research with well-defined tasks. For instance, the step that students explain in a well-defined task is a part of the algorithm, thus it required students to learn prior steps to be able to explain the prompted step. In an ill-defined task, students can explain steps without necessarily knowing other steps. For instance, in an ill-defined task such as writing an essay, students can explain ‘how to write a conclusion’ without knowing ‘how to write an introduction’. A limitation of our study is the small number of participants. It would therefore be interesting to see the results of a larger study. Moreover, it could be argued that this result is due to differences between conceptual-focused and procedural-focused self-explanation. As discussed previously, we used 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).

Sweller and Cooper (1985) explained a two-step learning process. First, examples are a suitable approach for students, particularly novices, since examples reduce the cognitive load and increase initial learning. Second, students use the knowledge they learnt from studying examples in solving similar problems. Our results are in line with two-step learning process. However, using an ITS instead of examples leads to 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 an ITS for novices, then when students’ knowledge increases, the system can continue giving them a mixture of examples and problem solving, or gradually switch to ITS only. The result suggests that for a long learning time, problems only may even out-perform the alternating examples and problems condition since advanced students do not need any more knowledge; what they need is to apply knowledge in solving new problems. Nevertheless, in our study, novices learnt the most from AEP since SQL is not completely novel to them (all the students in our study attended several lectures about SQL a week before the study). Thus, if we adapt the proportion of examples and problems to students’ needs, the

system may work more efficiently than EO, AEP and PO at any stage of learning. Therefore, we proposed and compared an adaptive model for using examples which is discussed in the next chapter.

Chapter 6. Second study: Adaptive examples in

SQL-Tutor

The results of the study described in this Chapter were published in (Shareghi Najar et al., 2014a).

Little attention has been devoted thus far to the difference between novices and advanced students in learning from examples and learning from supported problem solving. In the initial stages of learning, examples should be alternated with problems solving (Sweller & Cooper, 1985). When students gain enough knowledge, they should be provided problems only (Kalyuga et al., 2001). Research shows that students need different levels of guidance (Koedinger & Alevan, 2007). The amount of assistance should be adapted to students' needs. Although there have been a few attempts to adapt examples to the student's level of expertise (Salden et al., 2009a; Kalyuga & Sweller 2005), there is still no agreement on how much assistance should be provided to students.

In Chapter 5, we discussed 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. We scaffolded examples and problems with Self-Explanation (SE) prompts, requiring students to explain worked examples provided or how they solved problems. The results showed that students benefitted the most from alternating examples and problems. In that study, we used a fixed sequence of examples and problems; therefore, it is possible that some students have received less or more information than they needed. This encouraged us to propose a new adaptive learning strategy that decides what type of task to present to the learner. The learning tasks are problem solving, 2-step faded examples, 1-step faded examples and worked examples, with faded steps chosen based on the student's performance.

This chapter presents the study conducted to evaluate this adaptive strategy by comparing it to the best condition (AEP) from the study we discussed in Chapter 5. We start by describing the study, while Section 6.2 presents the results. We discuss results in Section 6.3.

6.1 Study

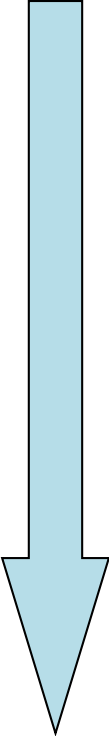
As in the study in Chapter 5, we used web-based SQL-Tutor (Mitrovic, 2003). The version of SQL-Tutor we used in this study had four modes: problem solving, 2-step faded example, 1-step faded example and worked example. The problem-solving mode is similar to the original SQL-Tutor in which students solve problems. The 2-step faded example mode and 1-step faded example mode are similar to the problem-solving mode in which the student needs to complete two or just one step (i.e. clauses of the SELECT statement). The fourth mode is the worked-example mode, which presents the complete solution and an explanation to the student. In all four modes students have access to the database schema at the bottom of the screen.

6.1.1 Experiment design

The study was conducted in a single, 100-minute long session. Figure 6.1 shows the design of the study. The participants were volunteers from an introductory database course at the University of Canterbury. The study was conducted during the tenth week of the semester. Students had learnt about SQL in lectures beforehand and needed to practise in the lab. The students did not receive any inducements for participating in the study, but we told them that working with our system would help them learn SQL. We informed them that they would see ten pairs of problems and that the tasks in each pair were similar. At the beginning of the session, the students took a pre-test (given in Appendix B.3) for ten minutes. The pre-test had ten questions, eight of which were multiple-choice and two were problem-solving questions. The multiple-choice questions (worth one mark each) measured conceptual knowledge (e.g. *what clause of the SELECT statement allows the resulting table to be sorted?*). For the problem-solving questions, students had to write SQL queries. Those two questions (worth four marks each) measured procedural knowledge and problem-solving skills (e.g. *write a query that retrieves the names of all departments located in Houston*).

Once students logged in, SQL-Tutor randomly allocated them to one of the conditions, giving sample sizes of 24 in the control group and 24 in the experimental group. The participants studied ten pairs of isomorphic tasks. The complexity of pairs gradually increased from Pair 1 to Pair 10.

The control condition worked with example-problem pairs: each pair consisted of an example followed by an isomorphic problem. The only exception is the first pair, in which the control group received a problem followed by an example; this was so that the first problem could provide the necessary information for the adaptive strategy. The control condition in this study is identical to the best condition (AEP - alternating examples/problems) from the first study (with the exception of the first pair).



	Control	Experimental
	n = 24	n = 24
	Pre-test	
Pair 1	1 st task in the pair: problem	1 st task in the pair: problem
	2 nd task in the pair: example	2 nd task in the pair: rehearsal task (problem, 2 or 1 step faded example, worked example, or skip)
Pair 2 to 10	1 st task in each pair: example	1 st task in each pair: preparation task (problem, 2 or 1 step faded example, worked example or skip)
	2 nd task in each pair: problem	2 nd task in each pair: problem
	Each problem followed by a C-SE prompt and each example followed by a P-SE prompt	Each problem or faded example followed by a C-SE prompt and each example followed by a P-SE prompt
	Post-test	

Figure 6.1 Design of the study

The experimental group had pairs consisting of a preparation task followed by a problem, except for the first pair. In the first pair, the experimental group received a problem followed by a rehearsal task. Rehearsal tasks are the same as preparation tasks, but because they were provided after the isomorphic problem we called them rehearsal tasks. Our adaptive strategy decided what type of preparation task to present to the experimental group. As we mentioned before, all students (the control group and the experimental group) received Conceptual-Focused Self-Explanation prompts after problems and Procedural-Focused Self-Explanation prompts after examples. Once students solved a problem and answered a C-SE prompt, the system asked them to rate their cognitive load.

At the end of the session, students were given ten minutes to complete the post-test (given in Appendix B.4). However, students could start the post-test during the learning session and finish the study earlier. The post-test was isomorphic to the pre-test.

The fading strategy is based on the student's performance on the current task. Domain knowledge is represented in SQL-Tutor as constraints. Every time the student submits an attempt, the system analyses it and records information about the constraints that were satisfied or violated. It is therefore possible to find out how much the student learnt about a particular domain concept by comparing his/her knowledge before and after the current problem. Our fading strategy sorts the concepts that the student learnt in the current problem and selects the concept the student learnt the most (or the top two concepts, if two steps are to be faded). Then the system fades one or two steps of the next problem. If the next problem does not include the selected concept(s), the strategy fades the next concept (or two) from the sorted list. The idea is to help students rehearse what they have just learnt.

6.1.2 Self-explanation

Similar to the first study (discussed in Chapter 5), we presented students with a SE prompt after worked examples and problems. Conceptual-Focused Self-explanation prompts (C-SE) and Procedural-Focused Self-explanation prompts (P-SE) are meta-cognitive prompts in which students reflect on concepts required to solve the problem or on procedural steps of learning materials. Students were given C-SE prompts after problems or faded examples and P-SE prompts after examples.

Figure 6.2 shows a screenshot of a situation when a student had finished reading an example. Next, the system showed a P-SE prompt, located on the right side of the screen. The student gave a wrong answer to the P-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 to the next task.

Example 9

Find the names of artists and instruments they played in 'Someone to watch over me' or 'Summertime'.

```
SELECT lname , fname, instrument
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');
```

Explanation

The IN predicate allows you to enumerate values to be used in a comparison to an attribute in the WHERE clause.

Which option is equivalent with this condition?

```
title IN ('Someone to watch over me','Summertime')
```

- A) title = 'Someone to watch over me'
- B) (title = 'Someone to watch over me' or title= 'Summertime')
- C) (title = 'Someone to watch over me' and title= 'Summertime')
- D) (or (title = 'Someone to watch over me', title= 'Summertime'))

Wrong - the IN predicate can be replaced with OR.

[Next](#)

Figure 6.2 Screenshot of an example page followed by P-SE

SQL-TUTOR [History](#) [Log Out](#)

Problem 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 `song.id=recording.song and
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`

GROUP BY

HAVING

ORDER BY

Well done! Now, please answer the following question:

How much effort did you invest to complete this task?

Lowest 1 2 3 4 5 6 7 8 9 Highest

You can make only one attempt [Answer](#)

Figure 6.3 Screenshot of rating the effort after problem solving

SQL-TUTOR [History](#) [Log Out](#)

Problem 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	song.id=recording.song and 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
GROUP BY	
HAVING	
ORDER BY	

What is the role of NOT IN predicate?

A) It allows you to specify tables.

B) NOT IN allows you to specify a condition on an attribute checking that the value of the attribute appears in the enumerated set of values.

C) NOT IN allows you to specify a condition on an attribute checking that the value of the attribute does not appear in the enumerated set of values.

D) NOT IN allows you to define attributes in the SELECT clause.

Well done.

[Next](#)

Figure 6.4 Screenshot of a problem-solving page followed by C-SE

Figure 6.3 shows a screenshot of a problem-solving task. In this situation, the student was asked to rate the effort they invested to complete the problem on a 9-point rating scale. Once the student rated their effort, the student was given a C-SE prompt. Figure 6.4 shows that the student gave a correct answer to the C-SE prompt and the system showed the positive feedback.

The ‘History’ button not only shows solved problems and examples, but also past SE prompts and correct answers.

6.1.3 Calculating assistance score

Our adaptive strategy is based on the measure of assistance the student received while solving a problem. Table 6.1 shows the scores (H_i) we assigned to each level (i) of feedback in SQL-Tutor. H_i represents the assistance score for feedback level i . Level 0 (H_0) presents minimum assistance (score = 1) and level 5 (H_5) shows the maximum assistance (score = 6).

Table 6.1 Assistance scores for different levels of help

Name	i	H_i
Positive/Negative	0	1
Error flag	1	2
Hint	2	3
Partial solution	3	4
All errors	4	5
Complete solution	5	6

The easiest way to calculate the assistance score is to sum up the assistance score of all requested help, as in Equation 6.1. In the equation, U_i is 1 if H_i was requested, otherwise U_i is 0. In SQL-Tutor, students can ask for a level of feedback several times; therefore, as shown Equation 6.2, the assistance scores of feedback messages are multiplied by the number of times they have been requested in a problem (n_i).

$$\text{Assistance score: } T = \sum_{i=0}^5 H_i * U_i \quad (6.1)$$

$$\text{Assistance score: } T = \sum_{i=0}^5 H_i n_i \quad (6.2)$$

When a student has seen a particular feedback message then requests it again, the message does not contain the same amount of new information; therefore, the assistance score should be less than Equation 6.2. For instance, when a student sees a complete solution to a problem, the next time s/he asks for the complete solution, the same solution will be shown. Therefore, we had to make the effect of n_i converge to a number. In this strategy we multiplied the assistance score for each level of feedback by the power two series of n , with n showing the number of requests for the level of feedback (see Equation 6.3). Power two series converges to two, as shown in Equation 6.4.

$$\text{Power two series } (n): Po(n) = \sum_{j=1}^n \frac{1}{2^{(j-1)}} \quad (6.3)$$

$$\lim_{n \rightarrow \infty} Po(n) \approx 2 \quad (6.4)$$

In Equation 6.5, we rewrite Equation 6.2 using Equation 6.3:

$$T = \sum_{i=0}^5 H_i Po(n_i) \quad (6.5)$$

While Equation 6.5 appears mathematically sound and correct, it does not take into account the student's behaviour after receiving feedback. For instance, the current formula shows that a student, who solved a problem by receiving $H_0 H_1 H_2$ (without getting a partial or complete solution), received the same or more information as student B who saw a complete solution (H_5) once. Moreover, it is important to distinguish between students who complete problems with minimum assistance and students who request the complete solution in the first attempt. One way is to change the scoring system we presented in Table 6.1. However, changing the scoring system does not help to distinguish between students who saw a complete solution in the first attempt and students who saw a complete solution after several attempts to solve the problem. For instance, students who see a complete solution after several incorrect attempts search for their mistakes when they see the complete solution. Moreover, seeing a complete solution

in the first attempt encourages students to copy the solution and leads to very shallow learning (Deeks, 2000).

In order to include the student behaviour in the assistance score formula, we introduced parameter B, which represents the average score of requested feedback levels (Equation 6.6). As an example, when a student requests H₁ three times followed by H₄, the value of B is 3.5. Parameter B indicates whether the student tends to use high or low levels of assistance; for instance, if B is 2.5, the student mostly uses low feedback levels, but when B is 4.5, the student uses high levels of feedback rather than low-level feedback to solve the problem.

$$\text{Student Behaviour: } B = \text{AVERAGE}(H_m),$$

$$m \text{ is the list of requested hint levels} \quad (6.6)$$

This information was not available in Equation 6.5. Having such information, we can design an appropriate coefficient, but would a linear coefficient be a suitable approach (Equation 6.7)? Equation 6.7 does not discriminate well between different levels of feedback. For instance, there is a small difference between B = 1, B = 2, B = 3 or B = 4. In fact, B = 4 shows that students used a partial or a complete solution to accomplish the task, while B = 3 shows that students definitely did not see a complete solution, but might have used a partial solution in conjunction with some other low assistance hints. Therefore, we should use different slopes for each behaviour. An appropriate function that accounts for this is shown in Equation 6.8.

$$T = B \sum_{i=0}^5 H_i P_o(n_i) \quad (6.7)$$

$$f(x) = \sin\left(\frac{\pi}{2}\left(\frac{x}{3} - 1\right)\right) + 1 \quad (6.8)$$

In order to make a greater difference between low-level and high-level assistance scores, in Equation 6.9 we use a power two of Equation 6.8. Since g(x) starts from zero, we incremented the formula to avoid a zero coefficient and obtain Equation 6.10. We also changed the name of the function to *Skewness slope*.

$$g(x) = \left(\sin\left(\frac{\pi}{2}\left(\frac{x}{3} - 1\right)\right) + 1\right)^2 \quad (6.9)$$

$$\text{Skewness slope: } K(x) = \left(\sin\left(\frac{\pi}{2}\left(\frac{x}{3} - 1\right)\right) + 1\right)^2 + 1 \quad (6.10)$$

Overall, from Equation 6.7 and Equation 6.10, we rewrite the assistance score formula and Equation 6.11 shows the final result.

$$T = K(B) \sum_{i=0}^5 H_i Po(n_i) \quad (6.11)$$

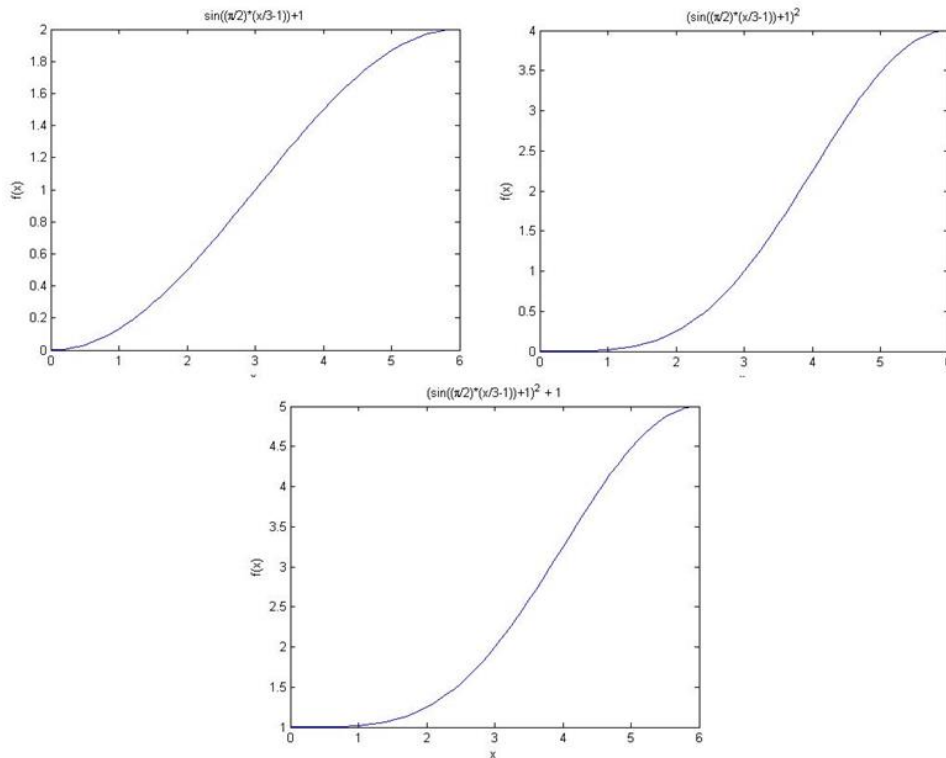


Figure 6.5 Plots of Equations 6.8, 6.9 and 6.10

We tested Equations 6.1, 6.5 and 6.11 using the data from our previous study, in which 12 students worked with SQL-Tutor. The students were given 20 problems whose complexities increased gradually. The students on average attempted 14.58 problems (sd = 5.11), resulting in 174 data points, of which 169 data points are from completed problems. Each data point represents a feedback level and the number of times feedback was requested by the student. We used 169 data points as an input dataset for our tests. The test was aimed to identify the accuracy levels of the three equations in separating data points in which only low-level feedback was requested (data points without H₃, H₄ and H₅). In the dataset we had 81 data points which contained only low-level feedback.

We identified the maximum possible assistance score when the data points did not contain high-level feedback (H_3 , H_4 and H_5). Table 6.2 shows the maximum assistance score achievable by using the three equations. Having the maximum scores, we could investigate how many data points fall within zero and the maximum score using each equation. The ideal equation should identify 81 points in the range between zero and maximum scores (100% accuracy). Table 6.3 shows that Equation 6.11 has the highest accuracy and therefore was used to calculate the assistance score after each problem was solved. Although the difference between Equations 1 and 11 is small, in developing Equation 11 we considered all critical scenarios that can fail Equation 1. Therefore, in sessions with more critical scenarios the difference would be more obvious.

Table 6.2 Maximum assistance score when students do not ask for high-level feedback

	H_0	H_1	H_2	H_3	H_4	H_5	Max Score
Equation 1	> 0	> 0	> 0	0	0	0	6
Equation 5	∞	∞	∞	0	0	0	≈ 12
Equation 11	0	∞	∞	0	0	0	≈ 15.5

Table 6.3 Result of the equations' evaluation

	Total samples between 0 and max score	Samples with low-level feedback	Accuracy (%)
Equation 1	95	81	85.2
Equation 5	106	81	76.4
Equation 11	93	81	87.1

6.1.4 Calculating cognitive efficiency

Paas and van Merriënboer (1993) calculated cognitive efficiency from the difference between the z-scores of performance (P) and mental effort rating (R), $CE = z_P - z_R$. In this way, CE can only be calculated after the experiment is completed. In order to indicate CE in real time, Kalyuga and Sweller (2005) used mental effort (R) and performance (P) to calculate Cognitive Efficiency as $CE = P \div R$. Mental effort was indicated by students and performance was calculated from the number of steps students required to solve a problem.

Our adaptive strategy is also based on a measure of cognitive efficiency. The participants were asked to rate mental effort (R) after solving each problem (i.e. *how*

much effort did you invest to complete this task?) on a 9-point rating scale. We calculated the student's performance, P from the assistance score, T :

$$P = T_{High} - T \quad (6.12)$$

When a student asks for a partial solution several times, effectively the student modifies the problem into a worked example. Examples provide maximum assistance; the assistance score for the situation when the student has seen partial solutions several times corresponds to a high level of assistance, which we refer to as T_{High} . Thus, using Equation 6.11 we calculate T_{High} to be 26 ($H3 = 4$; $K(4) = 3.25$). Therefore, performance (P) can be calculated as:

$$P = 26 - T \quad (6.13)$$

Note that T can have a value greater than T_{High} . Because T_{High} represents turning problems into examples, we set all assistance scores greater than T_{High} to 26. Therefore, P never becomes negative.

Performances are scaled to the range $[0, 9]$. Like Kalyuga and Sweller (2005), we defined the critical level of cognitive efficiency as $CE_{cr} = P_{max} \div R_{max}$, where P_{max} and $R_{max} = 9$. We consider $CE > CE_{cr}$ to be high cognitive efficiency; thus, students who solved a problem with $CE > 1$ were expected to be able to solve the next problem without needing a preparation task.

6.1.5 Adaptive strategy

The first pair of tasks is different from the other pairs. In this pair, the participants worked with problem 1 followed by a rehearsal task. A rehearsal task is the same as a preparation task, but because this preparation task is provided after problem 1, we refer to it as a rehearsal task. If the student's CE is greater than 1 in Problem 1, the system skipped the rehearsal task from the first pair and the preparation task of pair 2. As CE scores were updated after solving problems only, in the preparation task of the second pair the students received the same type of task as the rehearsal task from the first pair. The system behaviour for the second pair is the same as for all later pairs, as depicted in Figure 6.6.

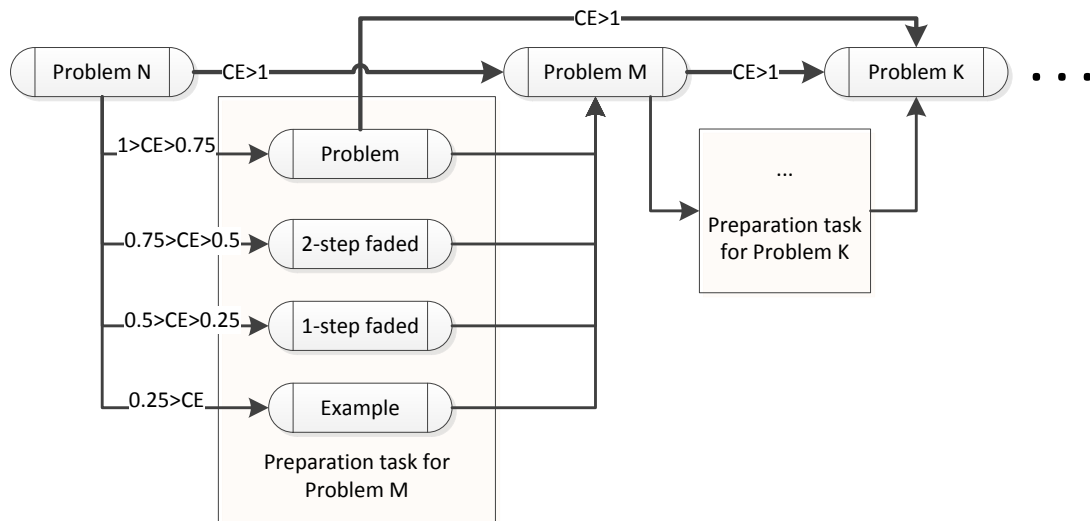


Figure 6.6 Study flow

Our adaptive strategy uses CE to decide whether or not the student needs preparation before the next problem, as shown in Table 6.4. A CE below 1 and above 0.75 (6.75/9) shows relatively good performance on the current problem, but indicates the need to prepare for the next problem by solving an isomorphic problem first. Students with CE between 0.75 (6.75/9) and 0.25 (2.25/9) received 2-step or 1-step faded examples as a preparation task. As we mentioned before, the steps are faded based on how much the student has learnt from the current task for each concept. Students who scored below 0.25 (2.25/9) get an isomorphic worked example before solving the next problem. When a student asked for a partial solution more than twice, or saw the complete solution, the strategy presented a worked example as a preparation task regardless of the student's CE. This is an exception to Table 6.4, because we think that when a student asks for a several partial solutions or an example, s/he does not have sufficient prior knowledge to solve the next problem, which is more difficult than the current problem. The behaviour of the adaptive strategy is illustrated in Figure 6.6. The system calculates the CE score only after problems. If a student performed well ($CE > 1$) in a problem which is shown as a preparation task, the system skipped the next problem and the preparation task for the subsequent problem.

Table 6.4 Decision table

Condition	$CE > 1$	$1 > CE > 0.75$	$0.75 < CE < 0.5$	$0.5 < CE < 0.25$	$CE < 0.25$
Preparation type	Skip preparation	Problem	2-step faded example	1-step faded example	Worked example

Figure 6.7 shows the different areas corresponding to each Preparation type. Students with $CE > 1$ skipped the next preparation task and worked on the problem from the next pair (PS_m). For CE scores below 1, students received a preparation task. We plotted the performance of the students in the first study (Chapter 5) against the difficulty levels that students rated at the end of the learning session. Figure 6.8 illustrates the results: green (Δ) without using H_3 H_4 H_5 ; blue (+) used H_3 more than two times or used H_4 ; red (\times) used H_5 ; black (O) used H_5 and H_4 . We calculated performance from Equations 6.11 and 6.13. Although we used the difficulty level students indicated at the end of the session, the plot gives us a distribution of points. The difficulty level scores are scaled to the range [0, 9].

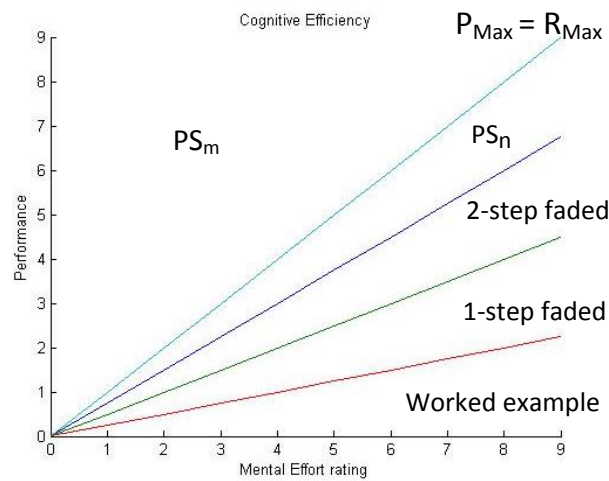


Figure 6.7 Diagram for cognitive efficiency and the model decisions

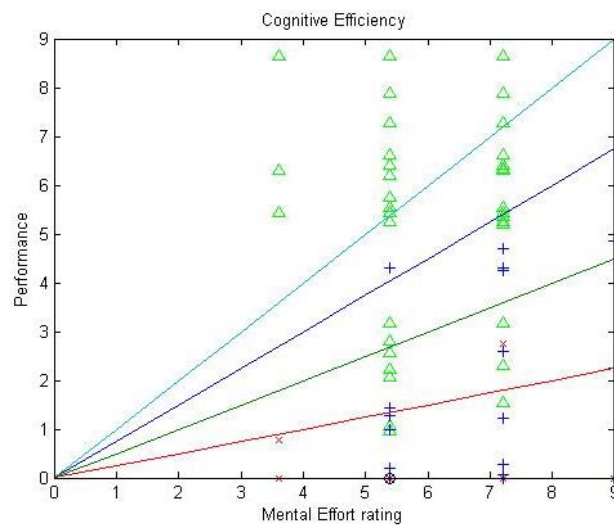


Figure 6.8 CE distribution from the previous study data

Although we used the cognitive efficiency proposed in (Kalyuga & Sweller, 2005), our approach is different from their model. We used the assistance score to measure performance while they used the number of steps students required to solve a problem (equation). Kalyuga and Sweller's model provides additional information for the current task, while we used the information from the current problem to predict the additional information students needed before working on the next problem. Our strategy chooses five different actions based on CE, while in the Kalyuga and Sweller model only four actions can be taken.

In the experimental condition, the amount of information students receive is adapted to the student's expertise level. Therefore, novices have a chance to study isomorphic examples when they do not know how to solve problems, while advanced students practise by solving problems only. The first study shows that novices benefitted the most from alternating examples and problems while advanced students benefitted the same from alternating examples and problems, and problems only. In the experimental condition, at the maximum assistance level the adaptive strategy becomes identical to the control condition and at the minimum assistance level it becomes the same as the problem-only condition from our previous study. Since the amount of information is adapted to the student's needs, our strategy will not cause the expertise reversal effect for the advanced students; thus our hypothesis is that the experimental group (especially advanced students) will learn more than students in the control condition. Moreover, the experimental group participants skip preparation tasks they have already mastered. Therefore, we hypothesise the experimental group will spend less time working with the system than the control group.

6.2 Results

Forty eight students from the University of Canterbury participated in this study. Two students from the control group were excluded from the analyses because they did not take the post-test. Thus, we had sample sizes of 22 in the control group and 24 in the experimental group. We calculated the average scores in the pre-test and the post-test and the time students spent on the system (Table 6.5). The students who had pre-test scores lower than 50% (median) were considered novices and the rest were classified as advanced students.

Table 6.5 Basic statistics for all students (standard deviation given in brackets)

Number of students	46
Pre-test (%)	48 (17)
Post-test (%)	82 (14)
Learning time (min)	66 (19)

We analysed the data to identify whether the two conditions affected learning differently and also to see whether the two conditions benefitted novices and advanced students differently. We start by explaining the results of the two conditions, followed by the results for novices and advanced students.

6.2.1 Analysing learning supported by the two conditions

The basic statistics of the two groups are presented in Table 6.6. There was no significant difference between the pre-test performances of the two groups. The t-test revealed a significant difference between the post-test results ($p = 0.05$). The performance of the control group was significantly lower than the experimental group, thus confirming one of our hypotheses. The students in both conditions improved significantly between the pre- and post-tests, as shown by the paired t-tests reported in the Improvement row of Table 6.6. Correlations between the pre- and post-test scores are also reported in Table 6.6, but only the control condition had a significant correlation ($p < 0.01$, $r = 0.55$). There was also a significant difference between the mean learning times of the two groups ($p < 0.01$). The experimental group spent significantly less time in the intervention than the control group.

The normalised learning gain⁹ of the experimental group was significantly higher than the gain of the control group ($p = 0.01$). When we analysed normalised learning gains on the conceptual knowledge questions (questions 1 to 8), we found no significant difference between the groups ($p = 0.13$). On the other hand, the normalised learning gain on procedural knowledge (questions 9 and 10) of the experimental group was significantly higher than the control group ($p < 0.1$).

⁹ Normalised learning gain = (Post test - Pre test) / (Max score - Pre test)

Table 6.6 Basic statistics for the two conditions (* denotes significance at the 0.05 level)

	Control (22)	Experimental (24)	p
Pre-test (%)	50.3 (13.7)	45.3 (18.9)	0.31
Post-test (%)	77.8 (13.9)	85.7 (12.6)	*0.05
Improvement	*p<0.01, t=-9.9	*p<0.01, t=-10.5	
Pre/post-test correlation	p<0.01, r=0.55	p=0.10, r=0.34	
Learning time (min)	73.6 (16.3)	58.9 (19.0)	*<0.01
Normalised learning gain (%)	55.7 (25.2)	73.2 (19.5)	*0.01
Conceptual knowledge gain (%)	76.4 (29.8)	87.7 (17.5)	0.13
Procedural knowledge gain (%)	29.5 (38.2)	62.0 (36.5)	*<0.01
Number of problems solved (inc. faded)	7.0 (2.5)	8.6 (3.0)	*0.06
Problems solved (excl. faded examples)	7.0 (2.5)	6.9 (2.4)	0.95
2-step faded		0.8 (1.2)	
1-step faded		0.9 (1.2)	
Number of examples	7.9 (3.0)	1.8 (1.9)	*<0.01
Number of attempts per problem	4.5 (2.0)	4.3 (1.7)	0.72
Maximum complexity level	13.4 (5.2)	14.0 (5.3)	0.71

Table 6.6 shows the experimental group participants solved marginally significantly more problems than the control group ($p = 0.06$), but the analysis involved faded examples. In order to solve faded examples, students had to fill in the faded steps. Therefore, we analysed the number of problems solved, excluding faded examples, and there was no significant difference between the two groups. We counted the number of 2-step and 1-step faded examples that the experimental group solved. The average number of 2-step faded examples solved by the experimental group was 0.8 and the average for 1-step faded examples was 0.9. The experimental group received significantly fewer examples than the control group ($p < 0.01$). There was no significant difference in the number of attempts per problem between the two conditions. The problem complexity gradually increased from pair 1 to pair 10. There was no significant difference between the average maximum complexity levels of problems the students in the two groups solved.

Students rated their mental effort after they solved problems (not after examples and faded examples as we could calculate CE scores after problems only), which the adaptive strategy used to calculate CE. As mental effort rate is specified on a 9-point scale, we used non-parametric tests for this analysis. First, we used Spearman's rho test to

investigate whether there is a correlation between the mental effort and the pre-test or between the cognitive efficiency and the pre-test.

Table 6.7 shows the results. We found a significant negative correlation between the pre-test scores and mental effort ratings ($p=0.03$, $r=-0.48$; $p=0.02$, $r=-0.48$, respectively) and significant correlations between the pre-test and cognitive efficiency in both groups ($p < 0.001$, $r = 0.69$; $p = 0.03$, $r = 0.44$). The table also shows significant negative correlations between the mental effort and cognitive efficiency in both groups ($p = 0.001$, $r = -0.67$; $p < 0.001$, $r = -0.73$). The significant negative correlations between mental effort and CE scores could be expected because CE scores were calculated from the mental effort. Next, we used the Mann-Whitney U test to compare the groups on CE and the mental effort. The results are summarised in Table 6.7. The table shows no significant difference between the experimental group and the control group ($p = 0.24$) on reported mental effort, but the experimental group had marginally significantly higher CE scores than the control group ($p = 0.09$).

Table 6.7 Cognitive efficiency and mental effort analysis

	Control (22)	Experimental (24)	p
Correlation: pre-test and mental effort	$p=0.03$, $r=-0.48$	$p=0.02$, $r=-0.48$	
Correlation: pre-test and CE	$p<0.001$, $r=0.69$	$p=0.03$, $r=0.44$	
Correlation: mental effort and CE	$p=0.001$, $r=-0.67$	$P<0.001$, $r=-0.73$	
Cognitive Efficiency (CE)	2.28 (2.29)	2.70 (1.85)	0.09
Mental effort	4.77 (1.71)	4.38 (1.20)	0.24

As mentioned earlier, the participants received C-SE prompts after problems and P-SE after examples. We analysed the SE success rates for the two groups, which are reported in Table 6.8. There was no significant difference between the groups in SE and P-SE success rates, but there was a marginally significant difference in C-SE success rate ($p = 0.08$). Students in the experimental condition performed better in C-SE than the control group.

Plotting learning curves investigates how the students in the control and experimental groups learnt SQL concepts in terms of constraints. For this, the number of times the constraints are relevant was plotted against the occasions when they were violated. A good fit to the power curve should result if the constraints evaluated are being learnt

(Martin & Mitrovic, 2005). Figure 6.9 shows the learning curves for the two groups. Both curves have a good fit to the power curve ($R^2 = 0.75$, $R^2 = 0.72$).

Table 6.8 Analyses of SE prompts

	Control (22)	Experimental (24)	p
SE Success rate	82.6 (12.2)	88.0 (12.5)	0.14
Procedural SE Success rate	90.3 (12.9)	90.0 (11.5)	0.97
Conceptual SE Success rate	73.6 (15.9)	84.0 (20.1)	0.08

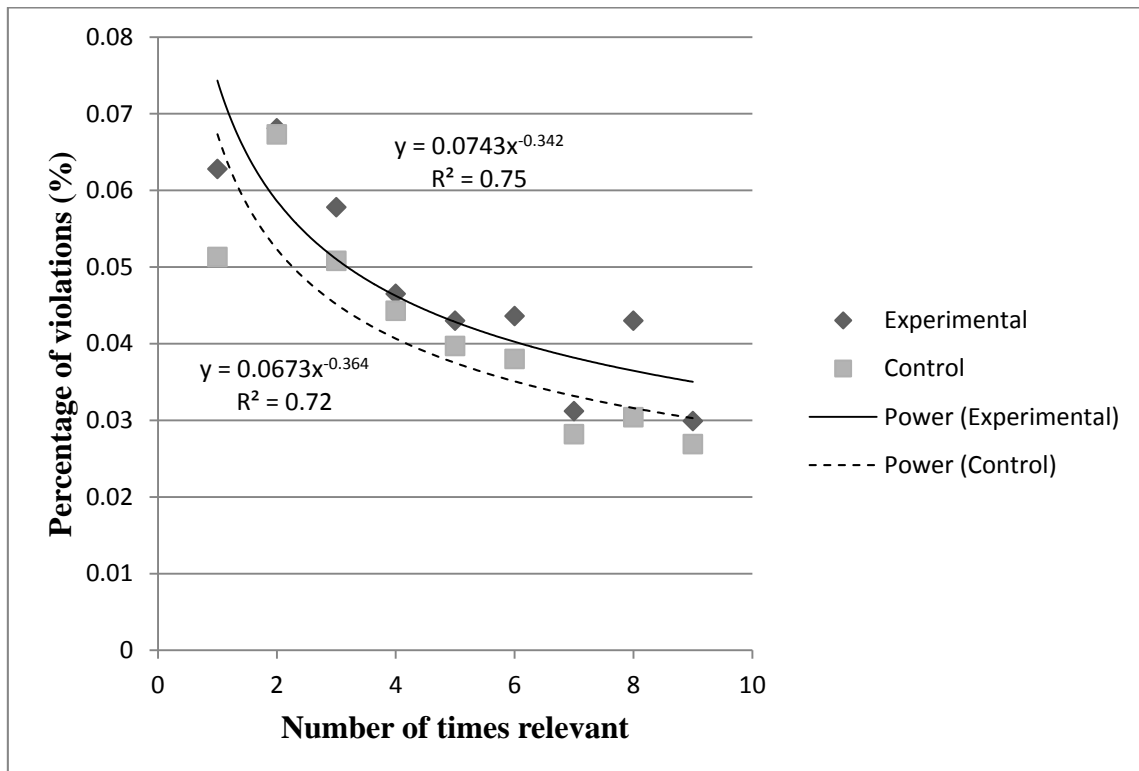


Figure 6.9 Learning curves for the experimental and control groups

Overall, the results show that the experimental group participants, who worked with the adaptive strategy, learnt more than students who worked with a fixed sequence of examples and problems. Moreover, the experimental group spent significantly less time working with the system than students in the control condition. The results clearly show the effectiveness of our adaptive strategy in comparison with the non-adaptive sequence.

Next we investigated how novices and advanced students benefitted from the control and experimental conditions. We calculated the effect size based on the normalised learning gain and on the cognitive efficiency scores using Cohen's d , with the following assumption: $d \geq 0.8$ (large effect), $d \geq 0.5$ (medium effect) and $d \geq 0.2$ (small effect)

(Cohen, 1988). The result is reported in Table 6.9. The effect size on normalised learning gain was medium and the effect size on cognitive efficiency was small.

Table 6.9 The effect size between the experimental group and the control group

Experimental and Control	Effect size (Cohen's d)
Normalized learning gain	0.75
Cognitive efficiency	0.21

In another analysis we were interested to find how each preparation task affected the cognitive efficiency score in the following problem. We extracted CE scores from the previous problems (CE₁) and from the following problems (CE₂) of each preparation task. This gave us 262 pairs of (CE₁, CE₂) from four types of preparation tasks in the adaptive condition and one type of preparation task in the control condition. Because of the low number of instances, 1-step and 2-step faded examples were considered as one type, named faded examples. We also excluded data from the first pair in which students had rehearsal tasks instead of preparation tasks. Note, in the skip action students did not see any preparation task; therefore, this condition is equal to not having preparation tasks. The result is summarised in Table 6.10. The table shows average CE scores before and after each preparation task. Figure 6.10 shows how average CE scores changed by having different types of preparation tasks.

Table 6.10 Summary of pairs

Condition	Action	Number of pairs	CE₁	CE₂	Improvement
Experimental	Example	35	0.03 (0.05)	0.90 (1.78)	*p < 0.01, t = -2.94
	Faded	15	0.50 (0.14)	2.06 (2.32)	*p = 0.02, t = -2.61
	Problem	9	0.90 (0.03)	2.88 (2.51)	*p = 0.04, t = -2.37
	Skip	73	4.01 (2.77)	2.32 (2.59)	*p < 0.01, t = 3.81
Control	Example	130	2.04 (2.37)	1.71 (2.21)	p = 0.14, t = 1.50

Next, we conducted paired t-tests to investigate whether or not CE scores significantly improved or deteriorated from problems given before preparation tasks to problems given after preparation tasks. The results are summarised in Table 6.10. The table shows that CE scores of students in the experimental group who had a preparation task significantly improved (example, p < 0.01; faded; p = 0.02, problem = 0.04). In the experimental group, CE scores of students who skipped preparation tasks significantly

deteriorated. This could be expected as students were not prepared for the next problem. However, their average CE scores is still above 1 (mean=2.32), which shows that students had enough knowledge to solve the next problem. This can be considered a trade-off between spending time on the preparation task that is not needed or skipping the preparation task and shortening the learning time.

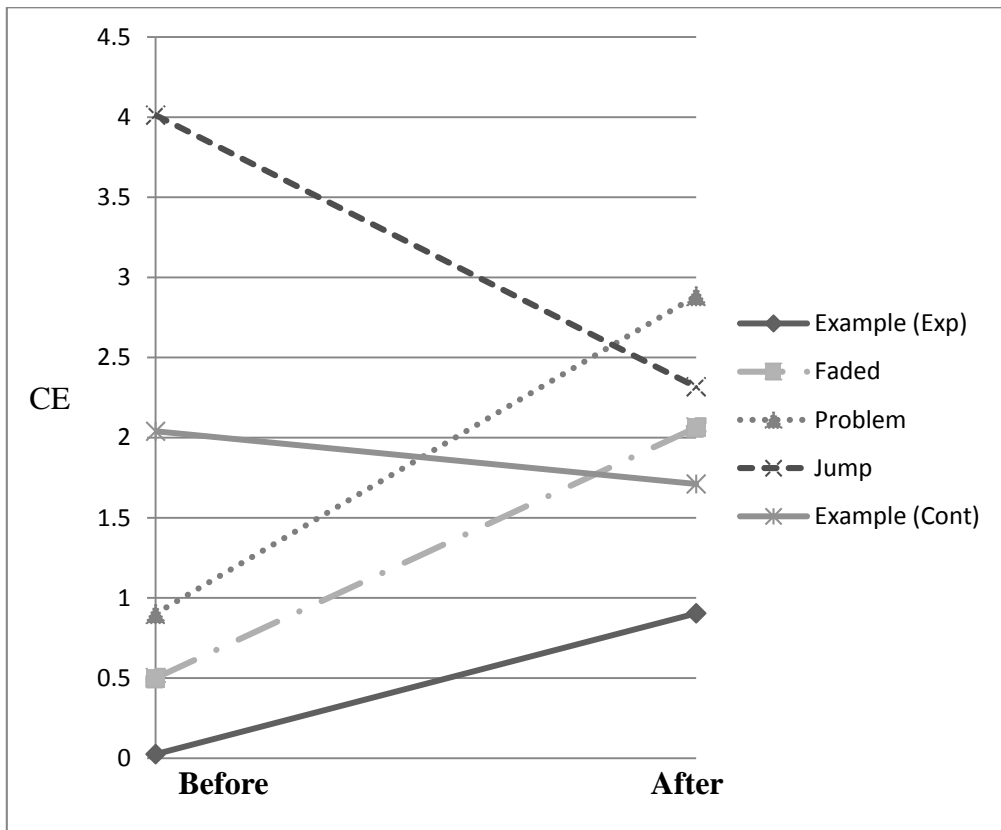


Figure 6.10 CE changes diagram

Table 6.10 shows that CE scores of the control group did not significantly change by studying examples; however, we see that CE scores of the experimental group, when examples were studied, significantly improved. This shows that the preparation tasks, particularly examples, provided were more targeted for the experimental group than the control group. Eventually, in the experimental group, students were given examples when their CE scores were below 0.25. That is, the students who received worked examples were novices, while in the control group all students saw an example before solving a problem. The average CE scores of the control group ($CE_1 = 2.04$ (2.37), $CE_2 = 1.71$ (2.21)) were bigger than the average CE scores of the experimental group ($CE_1 = 0.03$ (0.05), $CE_2 = 0.90$ (1.78)), because students who had examples in the experimental group

were all novices, while all students in the control group (including advanced and novices) had examples.

6.2.2 Results for novices and advanced students

We divided participants into novices and advanced students. Students who scored less than the median on the pre-test (50%) were considered as novices (control = 8, experimental = 13) and students who scored greater than median on the pre-test were considered advanced students (control = 14, experimental = 11). We used non-parametric tests for the following analyses.

The basic statistics for novices are shown in Table 6.11. The Mann-Whitney U test revealed no significant difference between the two groups on the pre-test and post-test scores. There was a marginally significant difference between learning times of the two groups ($p = 0.06$). The experimental group spent marginally significantly shorter time than students in the control group. The results show no significant difference between the two groups on the normalised learning gain, conceptual knowledge gain, procedural knowledge gain, number of problems solved (including faded examples) and problems solved (excluding faded examples). There was a significant difference between the numbers of examples the two groups studied ($p < 0.01$). The experimental group studied significantly fewer examples than the control group. The table also shows no significant difference between the attempts per problem for each group and no significant difference between the average maximum complexity level novices in the control and experimental groups saw.

Table 6.12 shows the results of cognitive load and cognitive efficiency analyses for novices. The Spearman's rho test revealed no significant correlations between the pre-test and mental effort and no significant correlations between the pre-test and cognitive efficiency among novices in the control group and experimental group. The Mann-Whitney U test was used to compare the experimental and control groups in CE and mental effort. The table shows novices in the experimental group experienced significantly higher CE than novices in the control group ($p = 0.02$). Moreover, novices in the experimental group indicated significantly lower mental effort than novices in the control group ($p = 0.05$).

Table 6.11 Basic statistics for novices

	Control (8)	Experimental (13)	p
Pre-test (%)	36.7 (8.5)	30.3 (8.4)	0.14
Post-test (%)	72.7 (15.6)	81.7 (14.5)	0.18
Learning time (min)	72.8 (12.5)	59.7 (16.0)	0.06
Normalised learning gain (%)	58.3 (21.7)	73.4 (20.4)	0.12
Conceptual knowledge gain (%)	85.7 (14.0)	94.6 (10.6)	0.19
Procedural knowledge gain (%)	32.8 (42.0)	55.4 (35.8)	0.24
Number of problems solved (inc. faded)	7.6 (2.1)	8.5 (3.0)	0.55
Problems solved (excluding faded)	7.6 (2.1)	6.6 (2.33)	0.34
2-step faded		0.77 (1.3)	
1-step faded		1.15 (1.3)	
Number of examples	8.87 (3.6)	2.3 (2.3)	*<0.01
Number of attempts per problem	4.6 (1.8)	4.6 (1.6)	0.75
Maximum complexity level	14.5 (4.8)	13.5 (5.4)	0.80

Table 6.12 Cognitive load and cognitive efficiency analyses for novices

	Control (8)	Experimental (13)	p
Correlation: pre-test and mental effort	p=0.60, r=-0.22	p=0.76, r=-0.09	
Correlation: pre-test and CE	p=0.21, r=0.50	p=0.28, r=0.33	
Cognitive Efficiency (CE)	1.04 (0.33)	2.34 (1.98)	*0.02
Mental effort	5.59 (1.04)	4.86 (0.87)	*0.05

Table 6.13 shows the result of the Mann-Whitney U test for SE success rate. We found no significant difference in SE, P-SE and C-SE success rates between novices in the control and experimental groups.

Table 6.13 SE prompts analyses for novices

	Control (8)	Experimental (13)	p
SE Success rate	79.0 (10.8)	87.4 (13.2)	0.16
Procedural SE Success rate	84.3 (9.1)	91.2 (10.1)	0.16
Conceptual SE Success rate	73.6 (13.5)	81.1 (20.1)	0.36

The basic statistics for advanced students are shown in Table 6.14. There are no significant differences between the pre-test scores and learning times. There was a significant difference in the post-test ($p = 0.02$), with the experimental group scoring higher than the control group. The normalised learning gain of the advanced students

from the experimental group is significantly higher compared to their peers from the control group ($p = 0.05$). There was no significant difference between the two groups in the conceptual knowledge gain, but the experimental group acquired more procedural knowledge than the control group ($p = 0.01$). The table shows no significant differences between the two groups in the number of problems solved (including faded examples) and problems solved (excluding faded examples). The control group participants had no faded examples and therefore studied significantly more examples than the experimental group. There is no significant difference between the numbers of attempts per problem, nor between the average maximum complexity level advanced students in the two groups received.

Table 6.14 Basic statistics for advanced students

	Control (14)	Experimental (11)	p
Pre-test (%)	58.0 (9.3)	63.1 (9.9)	0.17
Post-test (%)	80.8 (12.4)	90.3 (8.1)	*0.02
Learning time (min)	74.1 (18.5)	58.0 (22.8)	0.15
Normalised learning gain (%)	54.2 (27.7)	72.9 (19.5)	*0.05
Conceptual knowledge gain (%)	71.2 (35.2)	79.6 (20.9)	0.77
Procedural knowledge gain (%)	27.6 (37.3)	69.7 (37.5)	*0.01
Number of problems solved (inc faded)	6.6 (2.7)	8.6 (3.2)	0.12
Problems solved (excluding faded)	6.6 (2.7)	7.4 (2.5)	0.64
2-step faded		0.73 (1.0)	
1-step faded		0.55 (0.9)	
Number of examples	7.4 (2.7)	1.18 (1.17)	*<0.01
Number of attempts per problem	4.5 (2.1)	4.1 (1.8)	0.54
Maximum complexity level	12.8 (5.6)	14.6 (5.4)	0.50

Table 6.15 shows the results of cognitive load and cognitive efficiency analyses for advanced students. The Spearman's rho test revealed no significant correlation between the pre-test and mental effort for the control group, but there was a marginally significant negative correlation between the pre-test and mental effort for the experimental group ($p = 0.08$). There was a marginally significant correlation between the pre-test and cognitive efficiency among advanced students in the control group ($p = 0.07$), but no significant correlation among advanced students in the experimental group. The Mann-Whitney U

test shows no significant difference between the control and experimental groups in CE and mental effort.

Table 6.15 Cognitive load and cognitive efficiency analyses for advanced students

	Control (14)	Experimental (11)	p
Correlation: pre-test and mental effort	p=0.19, r=-0.37	p=0.08, r=-0.054	
Correlation: pre-test and CE	p=0.07, r=0.49	p=0.12, r=0.50	
Cognitive Efficiency (CE)	2.00 (2.64)	3.11 (1.67)	0.40
Mental effort	4.30 (1.87)	3.81 (1.31)	0.43

Table 6.16 shows the result of the Mann-Whitney U test for SE success rate. We found no significant difference in SE, P-SE and C-SE success rates between novices in the two groups.

Table 6.16 SE prompts analyses for advanced students

	Control (14)	Experimental (11)	p
SE success rate	84.6 (12.8)	88.8 (12.2)	0.43
Procedural SE success rate	93.7 (13.7)	89.0 (13.3)	0.24
Conceptual SE success rate	73.6 (17.8)	88.1 (20.9)	0.07

Overall, the results show that novices from the experimental condition learnt the same amount of knowledge as novices from the control group, but in a shorter time. Therefore, the adaptive strategy was more efficient for novices than studying a fixed sequence of examples and problems. Advanced students in the experimental group learnt more than advanced students in the control group while spending the same amount of time. Thus, the adaptive strategy is more efficient and effective than using a fixed sequence of examples and problem.

6.3 Discussion

We analysed the data from two perspectives: comparing the two conditions and analysing novices and advanced students separately. Students who worked with the adaptive strategy learnt more and faster than students in the non-adaptive condition. The participants in the experimental condition skipped the next problem if their cognitive efficiency was greater than 1, but students in the control group could not skip any tasks. There was no significant difference between the maximum complexities, which suggests

that students in both groups attempted the same problems. Therefore, we conclude that the experimental group learnt faster and acquired more knowledge than the control group.

We could not predict which group would learn faster before the study. The experimental group studied fewer examples than the control group, while from the first study (Chapter 5) we know that examples did not engage students like problems do; therefore, it was possible that the control group would spend less time working with the system than the experimental group.

Although post-test scores and normalised learning gains show that the experimental group learnt more than the control group, both groups acquired the same amount of conceptual knowledge. As students in both groups saw the same number of pairs, we could expect that the control group would learn more conceptual knowledge than the experimental group. Prior research shows that students learn more conceptual knowledge from studying examples than solving problems and students in the control group studied more examples than the experimental group. Therefore, no significant difference in conceptual knowledge reveals that our strategy has provided the right amount of conceptual knowledge for the experimental group. Moreover, the experimental group was more successful in answering C-SE prompts than the control group. C-SE prompts were only provided after problems and faded examples; as a result, the control group only saw C-SE prompts after problems. Therefore, a possible reason for a better C-SE success score by the experimental group is that the experimental group received faded examples before some C-SE prompts.

The control group acquired less procedural knowledge than the experimental group, while there was no significant difference between the two groups in the number of problems they solved (excluding faded examples). The experimental group had an adapted mixture of problems, faded examples and examples instead of examples only. Moreover, we faded solution steps adaptively. In faded examples, students worked on those parts of solutions that corresponded to recently learnt concepts. Therefore, not only did the adaptive learning strategy perform better than the fixed condition, but also applying the adaptive fading strategy improved students' procedural knowledge more than studying worked examples. The advantage of our adaptive strategy is also

corroborated by the CE analysis. The experimental group experienced marginally significantly higher cognitive efficiency than the control group.

We conducted paired t-tests to investigate whether or not CE scores had significantly changed after the preparation task. The results of the experimental group show that average CE scores significantly improved when students had examples, faded examples or problems. However, when students did not have a preparation task, the CE score significantly deteriorated. This could be expected because students were not prepared for the next problem. Nevertheless, the average CE score of such students was still above 1, which shows that their performance was above their cognitive load score. This was a trade-off in favour of learning time: students skipped the preparation tasks unless their CE scores dropped below 1. We believe this is the beauty of our adaptive model: it does not compel advanced students to do redundant tasks; consequently, the model avoids expertise reversal effect (Kalyuga, 2007). All students in the control group had examples as preparation tasks. The result shows CE scores of the control group did not significantly change; therefore, having examples in the control group was not as effective as having examples in the experimental group. The difference was whether or not the students had prior knowledge, not in the examples themselves which were the same. In the experimental group, students who had low prior knowledge were given examples to study, but in the control group, all students had examples.

Next we investigated how novices and advanced students benefitted from the proposed adaptive strategy. Novices in the experimental group acquired the same amount of knowledge, but did so faster than novices in the control group. The first study suggested AEP for novices and AEP and PO for advanced students. The results show that novices who worked with the adaptive strategy learnt the same as students in the control group, but in a shorter time. Moreover, novices in the experimental group had significantly higher CE scores than the control group and our strategy induced significantly less cognitive load than the control condition. The results clearly show that our strategy is a better option for novices than using a fixed sequence of examples/problems.

We did not see a significant difference in learning times for the advanced students of the two groups. We think that the time the experimental group saved due to skipping was

almost covered by the time that the control group saved by skipping studying examples or studying examples very quickly. Advanced students in the experimental group learnt more than advanced students in the control group. Perhaps advanced students in the control group suffered from an illusion of understanding or expertise reversal. C-SE success rate analysis reveals that advanced students in the experimental group were more successful than those in the control group. As the experimental group saw C-SE prompts after faded examples, not only after problems like the control group, we conclude that students learnt more conceptual knowledge from solving faded examples than solving problems. Therefore, our strategy worked for the advanced students.

When we look at the results for novices and advanced students, we can see that advanced students in the experimental group solved slightly more problems than the control group, while novices solved slightly fewer problems than the control group. Our strategy clearly shows that advanced students should be provided more problems than novices while, in a fixed sequence of examples and problems, both advanced students and novices had to solve the same number of problems and examples.

6.4 Conclusion

Prior research shows that students, particularly novices, learn more from examples than unsupported problem solving. While most of the studies that compared examples to ITSs indicate that students learn the same from worked examples and ITSs in domains with well-defined tasks, a few studies show that examples are not as effective as alternating examples and problems or problems only (e.g. (Kalyuga et al., 2001)). However, research shows that in the early stages of learning, using examples only or alternating examples and problems is superior than using problems only (Shareghi Najjar & Mitrovic, 2013b). This suggests that students benefit differently from worked examples and problem solving in different stages of learning. In the current study, we compared the best condition in the first study, a fixed sequence of examples and problems, with a novel adaptive strategy. First, we discussed a new approach to measuring assistance scores and consequently measuring performance scores. Using performance and mental effort rates, we could calculate the cognitive efficiency. In the adaptive strategy, we used cognitive efficiency scores to choose appropriate learning tasks for students. The strategy provided worked examples, faded examples and problems to students when they needed them. The fading

strategy was also adaptive: the system faded the solution steps related to the concepts that the student learnt the most in the most recent task.

In this study, a group of students worked with a fixed sequence of examples and problems (control group) and the other group worked with the adaptive strategy (experimental group). We explained the results from two points of view. First, we compared the two conditions, then compared novices and advanced students from the two groups. The results show that the experimental group learnt more and faster than the control group. For novices, the adaptive strategy led to faster learning, while for advanced students the adaptive strategy produced a higher learning gain than the non-adaptive model. Overall, the results prove that the adaptive strategy performs better than the non-adaptive one.

We explained a new formula to measure assistance. Using assistance scores, we can have a better understanding of how much assistance students should receive. We used assistance scores to calculate cognitive efficiency. Perhaps one of the other benefits of assistance scores is in identifying novices and advanced students while they solve problems. Note that students may have different knowledge levels for solving different problems. Knowing students' knowledge levels about a problem can help us to provide proactive feedback messages.

Prior research has shown that adaptive faded examples are superior to non-adaptive faded examples (Salden et al., 2009a), but their fading strategy was based on students' performance in answering self-explanation prompts. In our study, we used the student model to see how much students learnt about each concept, then faded the steps in the concepts students had learnt most recently.

Prior research also used cognitive efficiency to provide appropriate learning tasks (Kalyuga & Sweller, 2005), but they used students' performance which was based on how many steps students required to solve testing tasks (equations). In our study, we measured cognitive efficiency based on how much assistance students received when solving problems. Therefore, our strategy uses different features to make a decision.

Although we used self-explanation prompts in our study, in contrast with (Salden et al., 2009a), the adaptive strategy does not use self-explanation scores. The adaptive

strategy can be used for any ITSs that provide multi-level feedback, while the model proposed by (Kalyuga & Sweller, 2005) is designed for a specific domain (algebra).

Overall, the presented adaptive strategy improves learning more than the fixed sequence of examples and problems. In the first study (presented in Chapter 5), we showed that alternating examples and problems was superior to using examples only or problems only. Therefore, in this chapter we showed a further improvement in the form of adaptive presentation of problems and examples.

Chapter 7. Third study: Identifying learner differences in example processing

The results of the study described in this Chapter were published in (Shareghi Najari et al., 2014b).

The student's interaction with the ITS interface is of high importance, as it can reveal what students pay attention to. Eye-tracking data can be used to improve the student modelling and also provide adaptive supports (Kardan & Conati, 2012). There are also studies that investigate how students interpret various presentations of the Open Student Model by using eye trackers (e.g. Bull et al., 2007; Mathews et al., 2012). Eye tracking also enables investigations of successful and unsuccessful students' behaviour that lead students to learn or fail to learn. ITSs can classify students as novices or advanced students by matching their behaviour to successful and unsuccessful behaviours. We can leverage this information in two ways. First, the system can provide adaptive support, such as maximum or minimum assistance. Second, if the system knows that the student is a novice or advanced, the system can prompt the student to avoid unsuccessful behaviour and encourage productive behaviour (Kardan & Conati, 2012).

Most ITSs support problem solving, but research shows that incorporating examples into ITSs improves learning more than using problems only (McLaren & Isotani, 2011; Shareghi Najari & Mitrovic, 2013a; Shareghi Najari & Mitrovic, 2013b). As we explained in the previous chapters, in the early stages of learning learners benefit more from seeing worked-out examples (i.e. problems with solutions) than attempting to solve problems unaided. Therefore, investigating how novices and advanced students study worked examples will lead to new approaches for optimising learning from examples. However, so far researchers have mostly employed eye tracking to investigate how different students solve problems in ITSs. Examples (without scaffolding) do not have an interactive environment because students do not submit any response to the system (Koedinger & Alevan, 2007). Therefore, eye tracking can be leveraged to observe whether or not students study examples in a productive way. Moreover, it is interesting to observe how different students study worked examples. In order to get deeper insights

into how students use worked examples, in this chapter we report a study that used eye tracking to compare behaviour of novice and advanced students.

7.1 Experiment design

As discussed in Chapter 4, several studies have found the benefit of eye-tracking data in improving student models. In contrast, studies that investigate eye movement modelling examples (EMME) encouraged students to follow a successful searching behaviour; thus, students, regardless their knowledge level, were prevented from following unproductive approaches. To the best of our knowledge, prior research has never identified productive and unproductive behaviour while studying worked examples. Knowing such information could improve ITSs by prompting students to avoid unproductive behaviour and guide them towards successful behaviour. Therefore, we conducted a study to find productive and unproductive behaviour while students study SQL examples. We assume that novices show unsuccessful behaviour more than advanced students and that advanced students will study examples more productively than novices. Thus, it should be possible to identify productive and unproductive behaviour by comparing behaviours of novices and advanced students.

In this study we chose the Book database from 13 databases available in SQL-Tutor. Figure 7.1 shows the database schema; primary keys are underlined and foreign keys are in italics. We extended the system by adding the worked-example mode. In this study, we focus on how students study examples only.

AUTHOR	(<u>AUTHORID</u> , LNAME, FNAME)
PUBLISHER	(<u>CODE</u> , NAME, CITY)
BOOK	(<u>CODE</u> , TITLE, <i>PUBLISHER</i> , TYPE, PRICE, PAPERBACK)
WRITTEN_BY	(<i>BOOK</i> , <i>AUTHOR</i> , SEQUENCE)
INVENTORY	(<i>BOOK</i> , QUANTITY)

Figure 7.1 Book database schema

Figure 7.2 presents the screenshot of the worked-example mode, with a worked example at the top, followed by an explanation. The schema of the selected database is shown at the bottom of the screen. Once a student confirms that s/he has finished studying

the example (by clicking the button), the system presents a Procedural-focused Self-Explanation (P-SE) prompt.

For this study, we made several minor changes to the interface of the first study. We added fixed gaps (> 30 pixels for the 1920*1200 resolution) between the prompt text and each of the options in order to support identification of eye gazes. Moreover, we disabled scrolling to fix the position of page elements on the screen.

7.2 Participants and the procedure

We conducted the study using a Tobii TX300 eye-tracker (see Section 4.1) with a 23-inch monitor in the lab environment. An additional monitor was set up for an experimenter who monitored the eye-gaze movements in real time. Therefore, the experimenter could remind students to maintain their body position, in particular their head, if it caused eye gaze data to be lost.

Participants were the 22 students who had also participated in the first study (discussed in Chapter 5) in which we used a different database; thus, the participants did not see the same material. Each participant took part in the study separately and received a NZ\$20 voucher for participating. Initially, we asked students whether they had any vision problems and whether they needed glasses. Participants initially read an information sheet (Appendix C) and signed the consent form (Appendix D). After obtaining informed consent, we calibrated Tobii with the students' eye gaze. Tobii Studio™ provides a pre-calibration screen which shows whether or not the participant's eyes are being tracked. Figure 7.3 shows a screenshot of the pre-calibration screen. The experimenter then asked participants to sit in a comfortable position that maintained a green-coloured bar at the bottom of the screen. Then we started the main calibration phase as described in Section 4.1. The calibration was repeated if a poor calibration quality was seen.

SQL-TUTOR [Log Out](#)

Example 3

List the different cities that publishers are based in.

```
SELECT distinct city
FROM publisher;
```

Explanation

Some attributes in a table may contain duplicate values. However, you may want to list only different (distinct) values from a table. The DISTINCT keyword is used to return only distinct values.

What will happen if we don't use DISTINCT in this example?

- A) In that case all attributes will be selected.
- B) Only unique tuples will be selected.
- C) Then, the number of tuples may become larger than the number of cities.
- D) The system gives an error.

You can make only one attempt [Answer](#)

Schema for the BOOKS Database

The general description of the database is available [here](#). Clicking on the name of a table brings up the table details. Primary keys in the attribute list are underlined, foreign keys are in *italics*.

Table Name	Attribute List
AUTHOR	<u>authorid</u> lname fname
PUBLISHER	<u>code</u> name city
BOOK	<u>code</u> title <i>publisher</i> type price paperback
WRITTEN_BY	<u>book</u> <i>author</i> sequence
INVENTORY	<u>book</u> quantity

Figure 7.2 Screenshot of the worked-example mode of SQL-Tutor

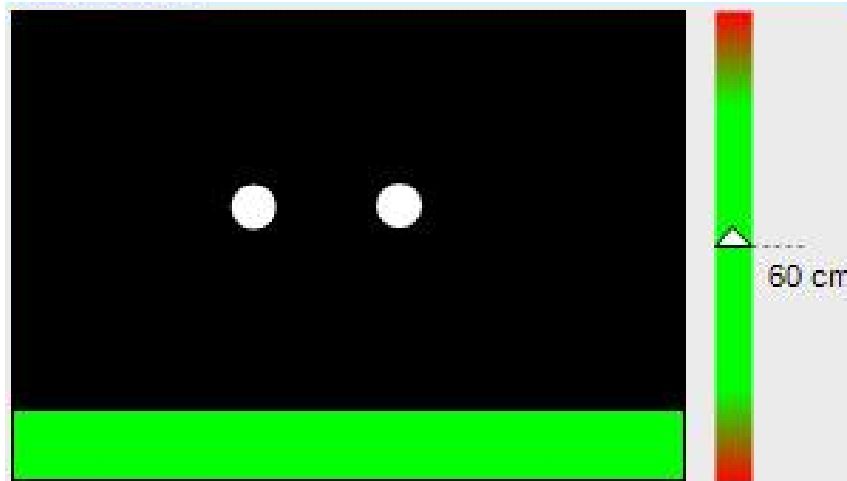


Figure 7.3 Pre-calibration screen

All participants used the same version of SQL-Tutor. SQL-Tutor was launched full screen within Tobii Studio™ software. The system presented six examples (Appendix A.2), each followed by a P-SE prompt. The students received examples and their explanations at the same time, and could make only one attempt at each P-SE prompt. When the student chose a wrong answer for a P-SE prompt, the system disclosed the correct answer and let the student continue using the ‘Next’ link. Each student was given a maximum of 50 minutes to study the examples.

All student actions were logged, including a timestamp, the example number, answers to the P-SE prompts, and when a student asked for additional information about the schema (such as attribute specifications). In addition to the logs, the eye tracker saved the screen video of each session; therefore, we had one recording per student.

7.3 Results

As the goal of the study was to observe how novices and advanced students study examples, we do not report eye-tracking data while students were answering P-SE prompts. We start by presenting the approach used to group students into novices and advanced students, followed by an analysis of time students spent studying examples.

7.3.1 Grouping the students

A common technique to identify novices and advanced students is to use pre-test scores, but in the current study we did not give a pre-test to the participants. The participants had

taken a pre-test in a study (Chapter 5) performed immediately before this study and we considered using those pre-test scores to divide the participants into novices and advanced students. However, the students worked with SQL-Tutor between the two studies; therefore, pre-test scores from the first study are not a precise measure of their incoming knowledge as they undertook additional learning since taking the pre-test.

For that reason, we used the K-Medoids algorithm (Kaufman & Rousseeuw, 1987) in order to cluster students into two groups. The inputs for the clustering algorithm were the pre- and post-test scores from our first study, and the P-SE scores and learning time from the current study. K-Medoids produced two clusters which we labelled novices and advanced, summarised in Table 7.1. The average scores of novices on the pre-test, post-test and P-SE prompts are lower than the average for the whole group (the Total column). The table also reports the total time that students spent on all examples (excluding the time spent on SE prompts), showing novices spent less time than advanced students for studying examples. There are significant differences between novices and advanced students on the pre-test, post-test and P-SE scores.

Table 7.1 Comparisons between the two clusters (standard deviations provided in brackets)

	Total (22)	Novices (12)	Advanced (10)	p
Pre-test (%)	40 (13)	33 (11)	48 (11)	<0.01*
Post-test (%)	70 (16)	63 (16)	79 (12)	0.02*
P-SE (%)	83 (13)	76 (11)	92 (9)	<0.01*
Time (s)	129 (54)	120 (59)	139 (52)	0.43

7.3.2 *Time spent studying examples*

Figure 7.4 shows that students spent noticeably more time on Example 1 than the following three examples, although the first example is the simplest one. We think this is because in Example 1 the students needed to learn the interface. That also explains why the time gradually reduced until Example 4. Afterwards, the time grew with increasing complexity of examples, except for the last two examples. Although Example 6 (nested queries) had a higher complexity than Example 5, Example 5 contained five clauses (SELECT, FROM, WHERE, GROUP BY, ORDER BY) while Example 6 only contained three clauses (SELECT, FROM, WHERE). Therefore, students needed more time to study Example 5 than Example 6. We compared the times novices and advanced students

spent on each example using the Mann-Whitney U test and found no significant differences.

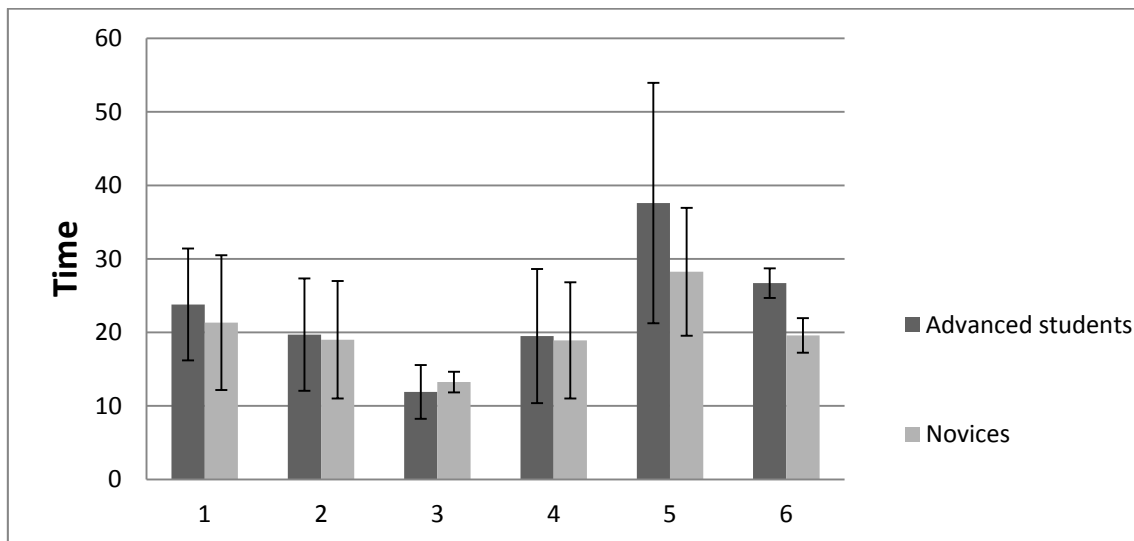


Figure 7.4 Average time (in seconds) spent on each example

7.3.3 Sample quality

Tobii studio provides an overall recording quality for the whole session, which is 77% for this study. We divided each recording into segments corresponding to individual examples. We calculated the quality for each segment by dividing the number of valid data samples (validity codes 0 and 1) by the total number of samples in the segment. The results are shown in Table 7.2. The data related to Participant 13 was excluded from further analyses due to technical issues and lost information.

The threshold for sample quality was set at 40% as we could not recognise patterns in recordings with qualities lower than 40%. The percentages in bold represent segments below the threshold; consequently, data for participants 5 and 10 were excluded from further analyses. Participant 16 studied examples after s/he saw the P-SE prompts, causing very short segments for examples and therefore we excluded data for this participant from further analyses.

Table 7.2 Quality of segments

Recording	Ex 1	Ex 2	Ex 3	Ex 4	Ex 5	Ex 6
1	89%	91%	83%	51%	84%	80%
2	92%	90%	90%	85%	90%	92%
3	93%	92%	92%	92%	91%	93%
4	69%	68%	78%	68%	69%	84%
5	5%	7%	2%	0%	6%	1%
6	95%	86%	93%	86%	87%	86%
7	100%	100%	100%	100%	97%	97%
8	96%	99%	99%	99%	99%	98%
9	99%	99%	99%	99%	99%	98%
10	73%	86%	62%	9%	29%	30%
11	79%	67%	57%	45%	44%	49%
12	99%	98%	98%	98%	97%	95%
14	98%	96%	97%	96%	97%	98%
15	95%	96%	99%	97%	95%	91%
16	100%	98%	100%	100%	100%	100%
17	97%	98%	99%	96%	95%	95%
18	95%	91%	90%	92%	91%	90%
19	63%	69%	83%	83%	90%	63%
20	93%	95%	94%	92%	72%	93%
21	83%	86%	85%	88%	84%	75%
22	93%	98%	100%	99%	100%	100%

7.3.4 Eye Gaze Pattern Analysis (EGPA) coding scheme

The goal of this study was to investigate how novices and advanced students study examples. In order to identify patterns of behaviour, we divided the interface into several Areas Of Interest (AOIs), as illustrated in Figure 7.5.

Area ‘W’ represents the worked example, ‘E’ is the explanation area and ‘D’ is the database schema. We then developed a new coding scheme named *Eye Gaze Pattern Analysis (EGPA)* to analyse the data. In EGPA, we define a number of patterns and behaviours. Patterns are those actions showing a student’s attention on an AOI or eye gaze movement from one AOI to another. A behaviour is a combination of patterns.

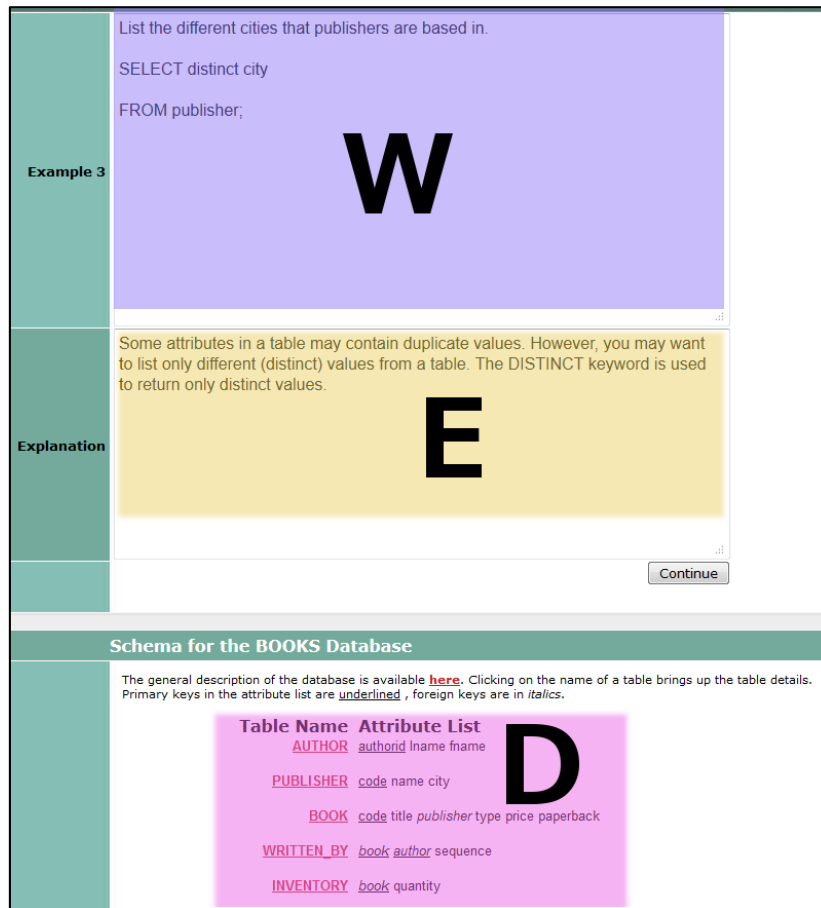


Figure 7.5 Areas of interest on the screen

Patterns are the smallest elements that explain eye gaze movements over a short time (1.5 s). We define four types of patterns: reading, mixed reading, transferring and scanning. A pure reading pattern indicates that a student is paying attention solely to one AOI. If the student glanced at an AOI while reading another area, we call it a mixed pattern. A transferring pattern shows that the student's eye gaze moved from one AOI to another. Finally, we define a scanning pattern, when the student scans the screen. Students scan the screen to learn the environment and once they finish scanning, they use one of the patterns above (e.g. reading pattern). EGPA contains the following types of patterns:

- X identifies that the student has only looked at area X. For example, seeing 'W' means that the student solely paid attention to the worked example. This pattern is reading.
- XyX identifies that the student had a short look at area Y while s/he was reviewing area X. For instance, 'EdE' shows that the student read an explanation

(E), but s/he had a quick look at the database schema (D) while s/he was reading the explanation. Mixed reading is this type of pattern.

- XY shows that the student's eye gaze moved from area X to area Y; for example, 'WE' means that student's attention changed from the worked example to the explanation. This type of pattern is transferring.
- S identifies that the student scanned the screen. This normally happens when a student sees the interface for the first time or when they are searching for information. Scanning is the type of this pattern.

As mentioned before, behaviours are a combination of patterns. An example of behaviour is W WE EdE EW WdW: the student first read the worked example, then the explanation. EdE shows that the student had a quick look at the database schema while s/he was reading the explanation. Then, the student's eye gaze moved from the explanation to the worked example and s/he had a quick look at the database schema while reading the worked example.

7.3.5 *Analysis of patterns*

Advanced students have higher prior knowledge compared to novices and might not need to read all the areas of interest. For that reason, we anticipated that advanced students would need fewer patterns than novices to study examples. Contrary to our expectation, the Mann-Whitney U test revealed no significant difference between the average pattern frequency used by novices and advanced students (Table 7.3).

We also used the Mann-Whitney U test to see whether there were significant differences between the two groups in pattern frequencies (Table 7.3). The advanced students used the D and ED patterns significantly and marginally significantly more often than the novices ($p = 0.03$ and $p = 0.08$, respectively). The D pattern was used by 90% of advanced students compared to only 25% of novices. The ED pattern was not used by novices at all, while half of advanced students used it.

Table 7.3 also shows that patterns EdE, WeW and EwE were used by 38, 63 and 75% of novices compared to 20, 40 and 60% of advanced students (respectively). There are different situations when the use of these patterns is productive. For instance, the explanations of examples 1, 2, 5 and 6 refer to attribute names, so in those cases it is

rational for students to look at the schema while studying explanations. Therefore, EdE is a productive pattern. The WeW and EwE patterns appear when the student is reading the worked example or the explanation and while doing that glances at the explanation or the worked example. This shows that the student is having difficulty processing the example or the explanation and has to remind him/herself of the other area.

Table 7.3 Average pattern frequencies

	Number of students		Average pattern frequency		
	Advanced (10)	Novices (8)	Advanced (10)	Novices (8)	p
All patterns			18.60 (5.19)	18.75 (5.26)	0.97
W	10 (100%)	8 (100%)	4.8 (2.2)	5.25 (1.39)	0.83
E	9 (90%)	8 (100%)	2.2 (1.32)	2.375 (1.19)	0.83
D	9 (90%)	2 (25%)	1.1 (0.57)	0.375 (0.74)	0.03*
WeW	4 (40%)	5 (63%)	1.2 (1.81)	0.625 (0.52)	0.90
WdW	5 (50%)	2 (25%)	1.4 (1.84)	0.25 (0.46)	0.24
EwE	6 (60%)	6 (75%)	1.2 (1.32)	2.125 (2.1)	0.41
EdE	2 (20%)	3 (38%)	0.3 (0.67)	0.5 (0.76)	0.57
WE	9 (90%)	8 (100%)	3.5 (2.01)	4.625 (1.69)	0.24
WD	4 (40%)	2 (25%)	0.4 (0.52)	0.25 (0.46)	0.63
EW	5 (50%)	4 (50%)	0.7 (0.82)	0.875 (1.13)	0.90
ED	5 (50%)	0 (0%)	0.5 (0.53)	0	0.08*
DW	3 (30%)	0 (0%)	0.3 (0.48)	0	0.32
DE	0 (0%)	2 (25%)	0	0.25 (0.46)	0.41
S	7 (70%)	8 (100%)	1 (0.94)	1.25 (0.46)	0.41

In contrast, WdW is a productive pattern since students need to check the database schema to understand an SQL example unless it was familiar from previous lab work. Table 7.3 shows that 50% of advanced students used WdW compared to 25% of novices. According to the DE pattern, 30% of advanced students checked the explanation after reviewing the database schema, while novices never looked at the worked example after studying the database schema. This shows that novices were not aware of the relations between the worked examples and the information available in the database schema. The scanning pattern (S) was used by 70% of advanced students while all novices scanned the interface. None of the novices knew where to start, when we showed them an SQL example.

7.3.6 Analysis of behaviours

Behaviour consists of an ordered list of patterns. We identified 42 distinct behaviours from 126 segments. Table 7.4 shows all the patterns in the segments. Behaviours that could not be identified are shown as 'N/A' and 'skip' shows that the student skipped reading the example. We first sorted behaviours by their frequencies; we refer to behaviours which appeared more than once as *frequent behaviours*. Frequent behaviours were sorted by the number of patterns and number of characters they contain, and finally their alphabetic order. Then we searched for the frequent behaviours within the non-frequent behaviours and changed the sequence of patterns that matched the frequent behaviour to the name of the behaviour (e.g. B3). For instance, 'WdW WeW' is a non-frequent behaviour consisting of two frequent behaviours (B9 and B8). As we searched in order, first B8 replaced 'WeW' and in the next round B9 replaced 'WdW'. Then we counted the number of times frequent behaviours appeared. We repeated the same procedure for novices and advanced students and computed the averages. The result is shown in Table 7.5. The Mann-Whitney U test showed no significant differences between novices and advanced students in behaviour frequencies.

Among the frequent behaviours, B3 was only used by advanced students. B3 shows a top down procedure in reading: the student first reads the worked example, then the explanation and finally the database schema. B3 is a logical routine that advanced students followed. As the advanced students had prior knowledge about the concepts covered in the examples, they looked at the explanation and database schema to find new information which they have not learnt before. Advanced students used B8, B9 and B10 more often than novices. These three behaviours contain only one pattern; therefore, advanced students used less complex behaviour than novices. Such simple behaviours may be explained by advanced students having more knowledge. On the other hand, novices used B2, B4 and B6 more than advanced students. In B2, students first studied the worked example followed by reading the explanation, and finally they restudied the worked example. B4 is similar to B2, but instead of restudying the worked example, students had a quick look at the worked example while reading the explanation. B6 represents that students first studied the worked example, followed by reviewing the explanation, but they did not pay attention to the database schema.

Table 7.4 Patterns recognised in each segment

Recording	Group	Example 1	Example 2	Example 3	Example 4	Example 5	Example 6	
3	Advance	S WdW WeW	WeW WE E S	S W	WeW WdW	D DW WeW WeW WE EwE	S	
4		W WE E	W WE E	W	W WE E	W	W	
6		S WdW	WdW	W WD D	W	W	WdW WdW	
7		W WdE EwE EW W WD D	WeW WeW	W WE E	W WE E	WeW	WE EwE	
9		S W WE E ED D	WeW EW EwE	W WE E EW W WE EwE	W WE E	W WE EwE	W WE EwE	
12		WeW WE EwE	W WE EdE ED D DW W	W WE E	W WE E EwE EW W	W WE EeE W	W WE EwE	
15		S W WE EwE	WeW WE E ED D	W WE E	WeW	WeW WdW WdW	WdW WdW WdW	
18		S W WE E EdE ED D	W WE E	W WE E	W WE S	W WE E	W WE E EW W	
19		S WdW WD D	W WE E	W WE E ED D	W	W	W	
22		S WdW WD D DW WE E	W WE W EwE EW WdW	W WE E	W	W WE EwE EW W	W	
1		Novice	W WE EwE	W WE E EW W	W WE E	W WE E	W WE EwE S	W WE E
2			S W WE E EwD D	WeW WE EwE EwE	WdW WD D DE E	W WE E EW W	W WE E	W WE EwE
8	S W WE EwE		W WE EwE	W WE EwE	W WE E	W WE EwE	W WE E	
11	S W WE EwE		W WE EwE EwE EwE EW W	W WE EwE	WE EW	W WE E	W WE EwE EW W	
14	S W WE EdE S		W WE E	W	W WE E	WeW	W WE E	
17	S W WE EdE		WeW S	W WE E	W WE E	WE EwE	WE	
20	S E EW W WE EdE EdE		W WE E EW W	W WD D DE E	W	WeW W	W	
21	WdW S		W	W WE E	W	WeW	W	
16	S W WD D		Skipped	Skipped	Skipped	Skipped	Skipped	
10	S E		S WE EwE EwE	W WE E	N/A	N/A	N/A	
5	N/A		N/A	N/A	N/A	N/A	N/A	

Table 7.5 Average behaviour frequencies

Name	Behaviour	Advanced	Novices
B1	W WE EwE EW W	0.1	0.1
B2	W WE E EW W	0.2	0.4
B3	W WE E ED D	0.2	0.0
B4	W WE EwE	0.4	1.3
B5	W WE EdE	0.1	0.4
B6	W WE E	1.5	1.8
B7	W WD D	0.2	0.1
B8	WeW	1.1	0.6
B9	WdW	1.0	0.3
B10	W	1.7	1.0

7.3.7 Analyses of eye-tracking data

In order to observe how often advanced students and novices looked at AOIs, we extracted the following metrics from Tobii Studio:

- Fixation duration (seconds): duration of each individual fixation within an AOI.
- Total fixation duration (seconds): duration of all fixations within an AOI.
- Fixation count (seconds): the number of fixations within an AOI.
- Visit duration (seconds): duration of each individual visit in an AOI.
- Total visit duration (seconds): duration of all visits within an AOI.
- Visit count (seconds): The number of visits with an AOI.

We used the Mann-Whitney U test to compare novices and advanced students for the above metrics within the three AOIs: the database schema (D), worked example (W) and explanation (E). Table 7.6 shows the results for D_{AOI} . There is no significant difference ($p = 0.14$) between novices and advanced students on the fixation duration. There are significant differences between the two groups on mean total fixation duration, fixation count, visit duration and total visit duration ($p = 0.03$, $p = 0.02$, $p = 0.01$, $p = 0.03$, respectively). That is, advanced students fixated more than novices on D_{AOI} ; moreover, total visit duration shows that advanced students spent significantly more time studying the database schema than novices. This result corroborates our previous analyses. Since

looking at the database schema is necessary to comprehend SQL examples, novices did not know how to study the examples.

Table 7.6 Eye-gaze metrics for novices and advanced students in D_{AOI}

	Novices	Advanced	p
Mean fixation duration (SD)	0.34 (0.16)	0.52 (0.24)	0.14
Mean total fixation duration (SD)	1.51 (1.52)	5.67 (5.23)	0.03*
Mean fixation count (SD)	6.75 (4.68)	27.50 (25.51)	0.02*
Mean visit duration (SD)	0.61 (0.38)	2.29 (1.59)	0.01*
Mean total visit duration (SD)	1.60 (1.59)	6.79 (6.73)	0.03*
Mean visit count (SD)	3.75 (1.91)	6 (3.97)	0.20

Tables 7.7 and 7.8 summarise the results for the W_{AOI} and E_{AOI}. There were no significant differences between the distributions of those metrics, showing that novices and advanced students both were aware of the importance of studying W_{AOI} and E_{AOI}.

Table 7.7 Eye-gaze metrics for novices and advanced students in W_{AOI}

	Novices	Advanced	p
Mean fixation duration (SD)	1.44 (1.63)	1.42 (1.80)	0.63
Mean total fixation duration (SD)	56.10 (15.41)	70.00 (40.33)	0.89
Mean fixation count (SD)	230.75 (46.03)	282.40 (146.21)	0.76
Mean visit duration (SD)	21.98 (9.49)	17.89 (9.47)	0.36
Mean total visit duration (SD)	68.80 (21.47)	82.01 (45.88)	0.96
Mean visit count (SD)	24.38 (8.93)	32.60 (14.84)	0.15

Table 7.8 Eye-gaze metrics for novices and advanced students in E_{AOI}

	Novices	Advanced	p
Mean fixation duration (SD)	1.19 (0.17)	1.17 (0.21)	0.90
Mean total fixation duration (SD)	28.30 (17.81)	26.82 (22.16)	0.63
Mean fixation count (SD)	130.25 (73.96)	117.50 (85.20)	0.79
Mean visit duration (SD)	10.82 (7.71)	8.43 (6.40)	0.46
Mean total visit duration (SD)	32.91 (20.50)	30.43 (24.69)	0.69
Mean visit count (SD)	19.75 (7.44)	20.40 (6.57)	0.76

7.3.8 Using machine learning classifiers

We also experimented with several Machine Learning algorithms in order to generate classifiers that predict the class of the student (advanced or novice) based on the patterns

and behaviours he/she exhibited while studying examples. In order to generate classifiers, we used RapidMiner 5 (Mierswa et al., 2006) with the WEKA plug-in (Hall et al., 2009) and selected the following algorithms: W-J48, W-Ladtree, W-BFTree, Rule Induction, W-JRip and Naïve Bayes. Most of the selected algorithms produce decision trees or rules and as such outputs are easy to interpret and visualise. The input vectors were specified in terms of 26 features listed in Table 7.9 (16 patterns and 10 behaviours).

The classifiers predict the class label (novice or advanced) using the whole session eye-movement data. Leave-One-Out Cross-Validation (LOOCV) is carried out on the normalised data and the results are given in Table 7.10. W-J48 performed the worst, with accuracy of 61.1%. On the other hand, W-Ladtree has the highest accuracy of 94.4%.

Table 7.9 Features used

Feature		
WE	E	B3
WeW	DW	B4
WD	DwD	B5
WdW	DE	B6
W	DeD	B7
EW	D	B8
EwE	S	B9
ED	B1	B10
EdE	B2	

Table 7.10 Accuracy of classifiers

Classifiers	Accuracy
W-j48	61.1%
Rule Induction	66.6%
W-BFTree	72.2%
W-JRip	77.8%
Naive Bayes	77.8%
W-Ladtree	94.4%

Figure 7.6 shows the W-Ladtree classifier, which predicted advanced students with 100% accuracy, while novices were predicted with 90% accuracy.

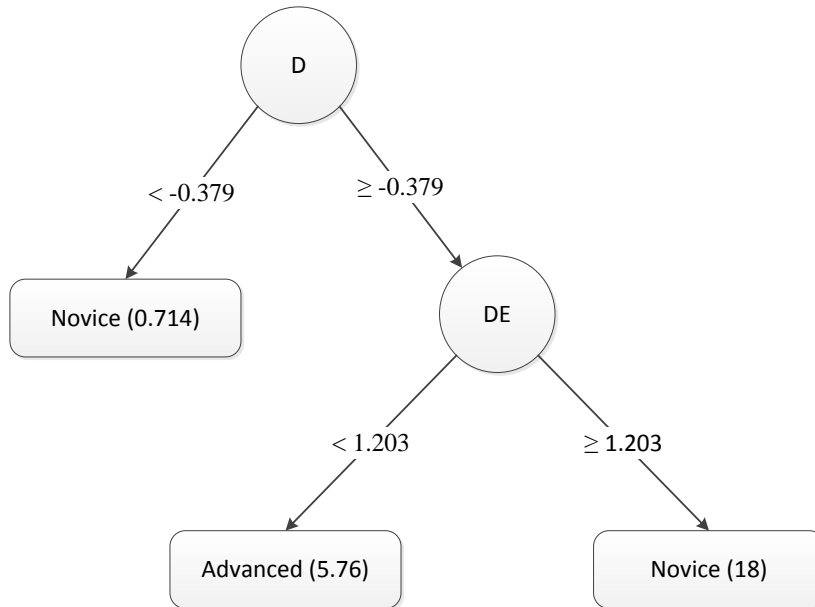


Figure 7.6 The W-Ladtree classifier

W-BFTree classifier does not predict novices accurately (50%), but does predict advanced students with 90% accuracy. The model is shown in Figure 7.7 and reveals that most advanced students spent more time on the database schema than novices.

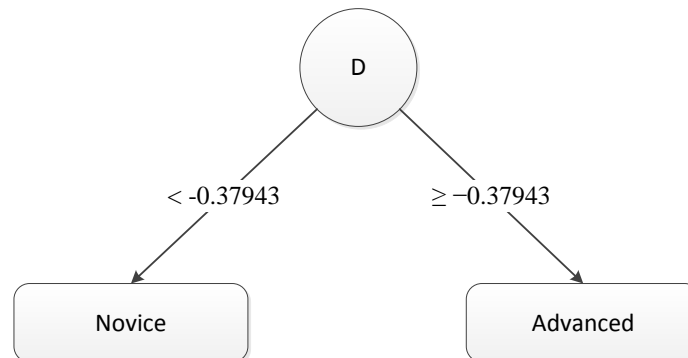


Figure 7.7 W-BFTree

The Rule Induction classifier (Figure 7.8) had 75.5% accuracy for novices and 60% accuracy for advanced students.

W-JRip achieved 77.8% overall accuracy; 80% in predicting advanced students and 75% for novices (Figure 7.9).

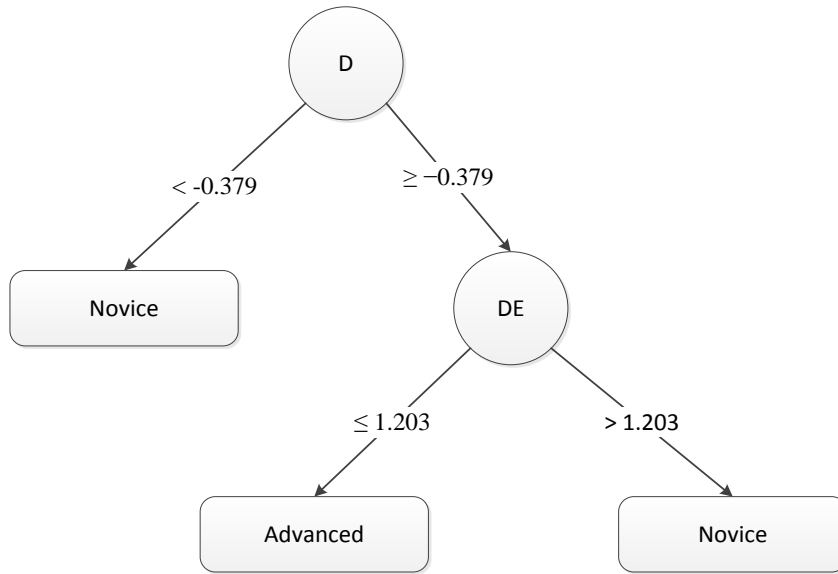


Figure 7.8 Rule Induction

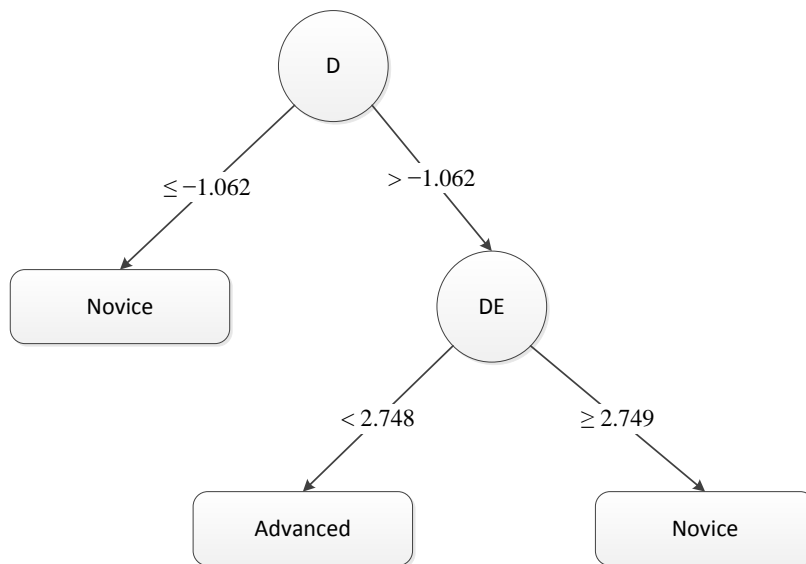


Figure 7.9 W-JRip

Overall, four classifiers (W-Ladtree, W-BFTree, Rule Induction and W-JRip) with high accuracies indicated that advanced students studied the database schema more than novices. Moreover, some novices looked at the explanation area after they studied the database schema. With a low reliability, W-j48 also indicates that five advanced students looked at the database schema more than the other students, and from the remaining students, three advanced students looked at the database schema while studying worked examples.

7.3.9 Heat maps

A heat map provides a graphical representation of data on a screen, in which colours are used to show data values (Bojko, 2009). In our study, data is the number of fixations and durations are relative to the total time participants spent visually attending the interface. Figure 7.10 shows the heat maps for Examples 1 and 2. The heat maps for Examples 3 and 4 are shown in Figure 7.11. Figure 7.12 illustrates the heat maps for Examples 5 and 6. The heat maps are from data for all students.

In Examples 1 and 2, students paid more attention to the explanation AOI than the latter examples. Students paid less attention to the explanation AOI in Examples 3 and 4 than Examples 5 and 6. The explanation changed from one example to another example. Students did not need to spend time reading the explanation, particularly when they knew the explanation would be about the material covered in the worked example AOI. Similarly, students paid more attention to the database schema during Example 1. Because the database schema did not change during the study, students did not need to check this AOI again. However, in Example 6, students paid attention to the database schema, but not as much as they did in Example 1. Students did not check the database schema in Example 4.

The heat maps show when students read the database schema in Example 1, they inspected most tables and their attributes. However, Example 1 is about the BOOK table (see Appendix A.2), so we expected students to pay more attention to the BOOK table and its attributes. However, the heat map shows that in Example 1 students paid attention to the whole database schema, not just the BOOK table. Example 6 shows the opposite behaviour. This example is about the BOOK and PUBLISHER tables and the heat map for Example 6 shows that students paid attention to the relevant areas in the database schema.

Overall, the heat maps show that students paid more attention to the AOIs in Example 1 than in the other examples. A simple reason is that students were not familiar with the environment; therefore, they tried to read all the information on the screen.

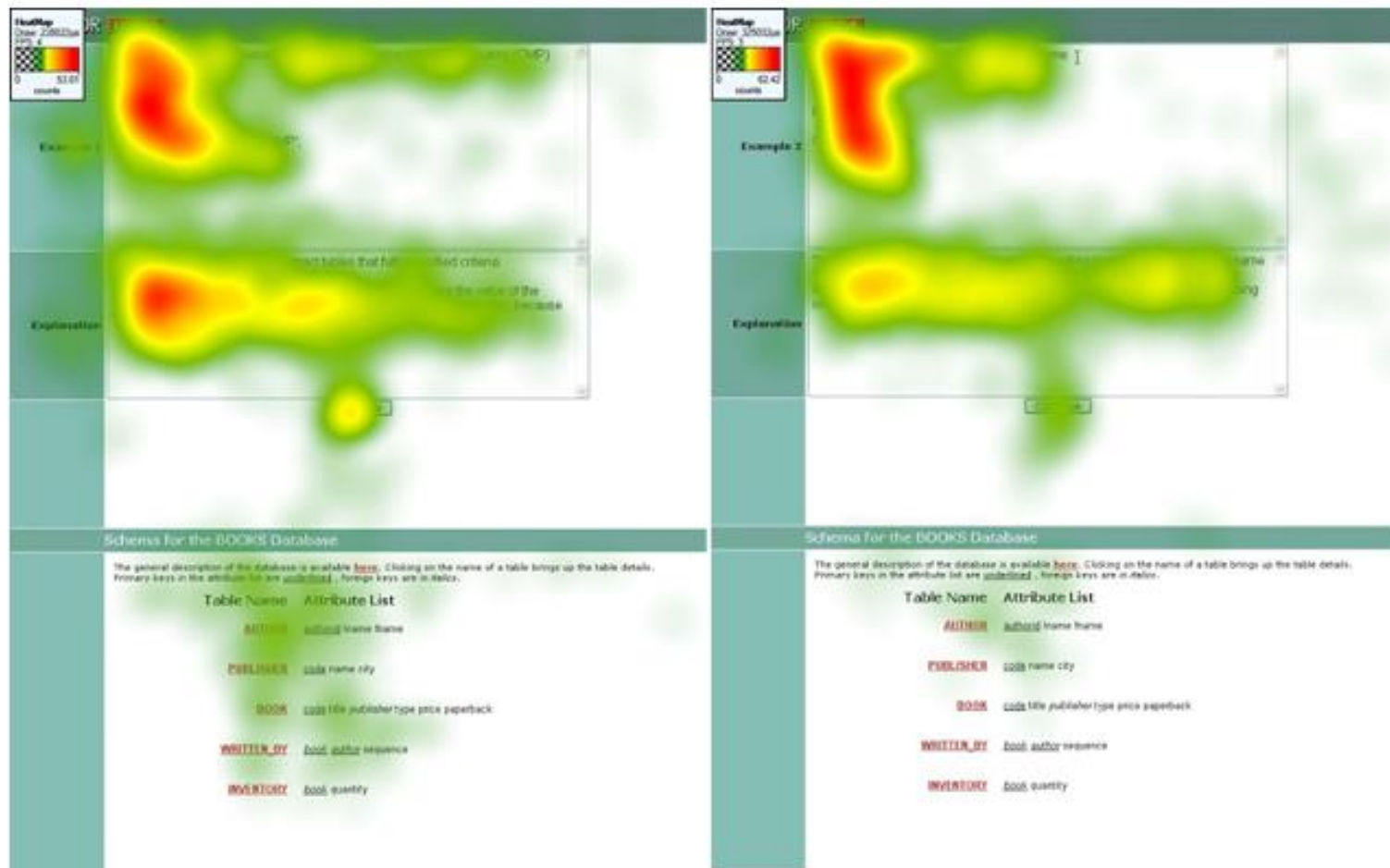


Figure 7.10 Heat maps for Example 1 (left) and Example 2 (right)

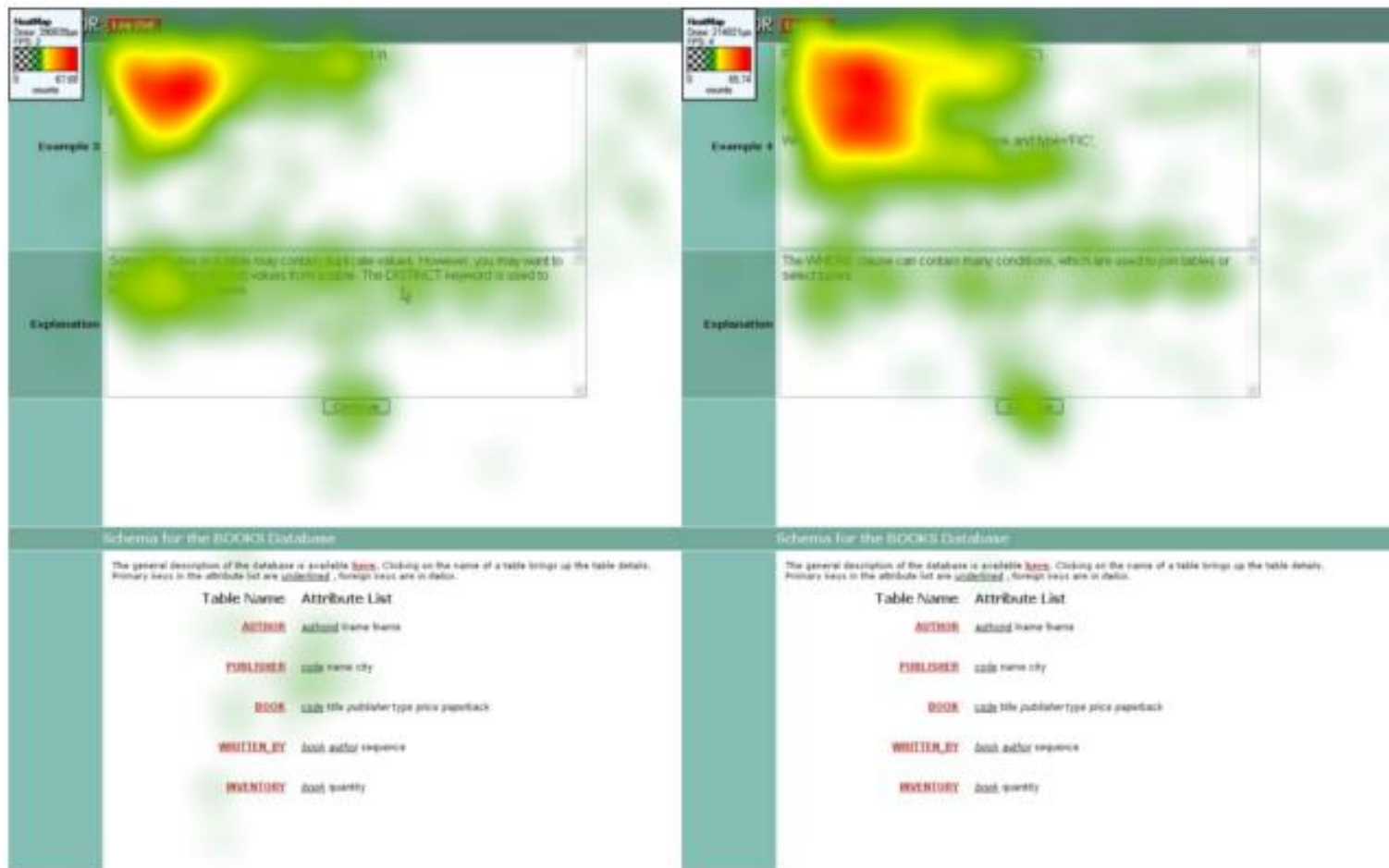


Figure 7.11 Heat maps for Example 3 (left) and Example 4 (right)

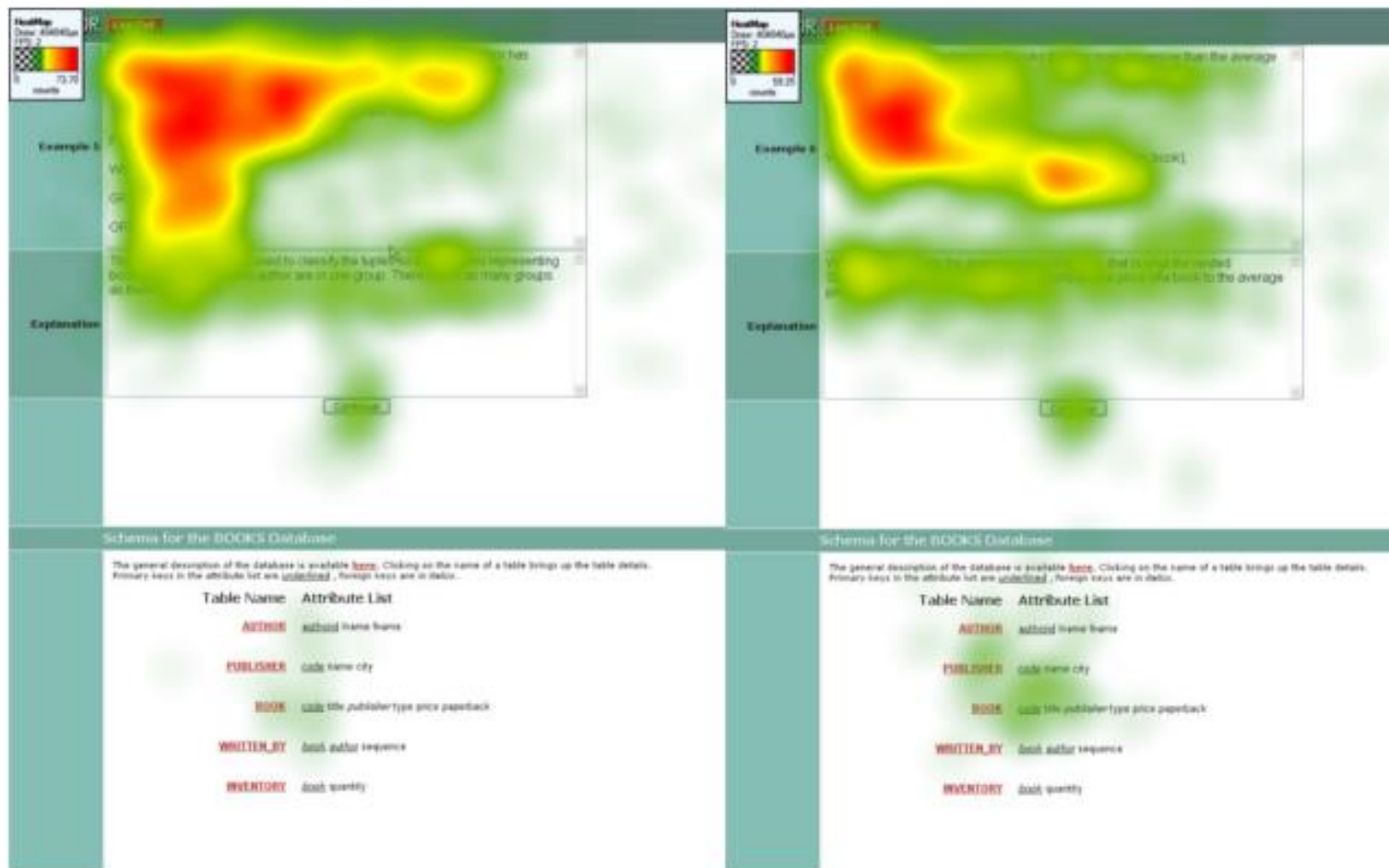


Figure 7.12 Heat maps for Example 5 (left) and Example 6 (right)

7.3.10 Gaze plots

Figures 7.13 and 7.14 show gaze patterns of a typical advanced student and a typical novice student on Example 1. The advanced student had a higher number of fixations. The figure shows that the advanced student had a comprehensive look at different AOIs while the novice student did not pay attention to the AOIs as much. The patterns clearly show that the advanced student read the problem statement then studied the solution. From the sequence of patterns we can see that the advanced student looked at the database schema when s/he was reading the problem statement. The information in the database schema helps learners to comprehend worked examples. The advanced student read all the tables and their attributes; this helped the student to know the primary keys, foreign keys and all information necessary for writing a query in SQL. The fixation numbers show that the advanced student read the explanation last. The patterns for the novice student do not imply whether or not the student has completely read the example or the explanation, or whether s/he had completely read the database schema. For instance, Example 1 is about the BOOK table, but the novice student did not fixate on any information about the BOOK table in the database schema.

In Example 1, apart from studying examples, students also tried to get familiar with the interface. Figures 7.15 and 7.16 show gaze patterns for a typical advanced student and a typical novice student on Example 6. The advanced student paid comprehensive attention to the problem statement and the solution. Although the student did not pay attention to the explanation, s/he had paid attention to the related areas in the database schema (BOOK and PUBLISHER tables). On the contrary, the novice student only looked at the worked example AOI. S/he did not have any fixation on the other AOIs. Even when the novice student studied the worked example, s/he did not pay attention to the AOI of the worked example as much as the advanced student did. There are three explanations that we identified for the novice behaviour on Example 6. First, the novice had learnt the concept before, so did not need to restudy it. Second, the novice went over the worked example very quickly, although s/he had not learnt enough. Third, the student did not know how to study an example of an SQL query; however, the student might know that s/he did not learn enough from studying the worked example. The results of this study and the study presented in Chapter 5 support the second and third explanations.



Figure 7.13 Gaze pattern for an advanced student on Example 1

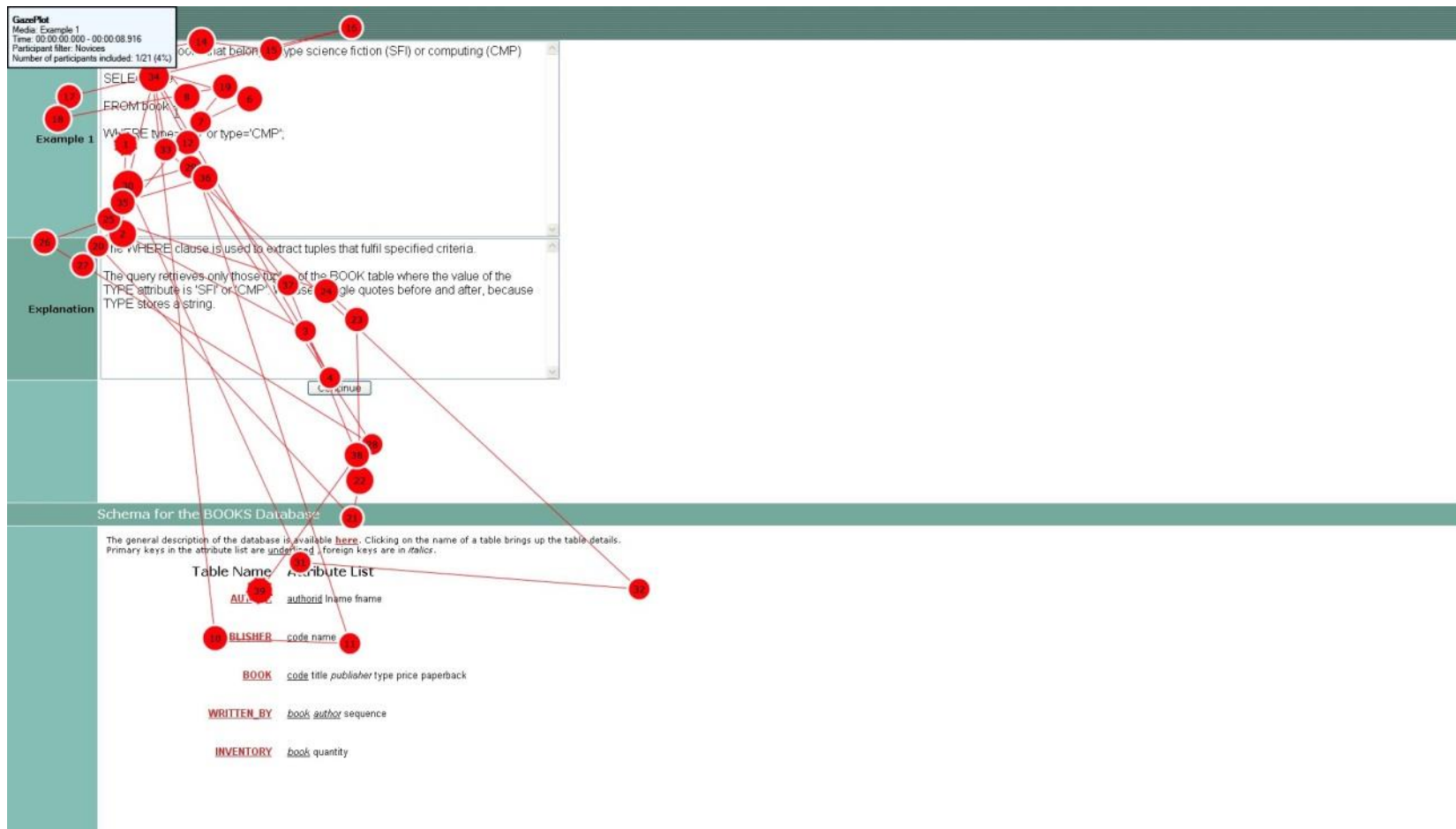


Figure 7.14 Gaze pattern for a novice student on Example 1

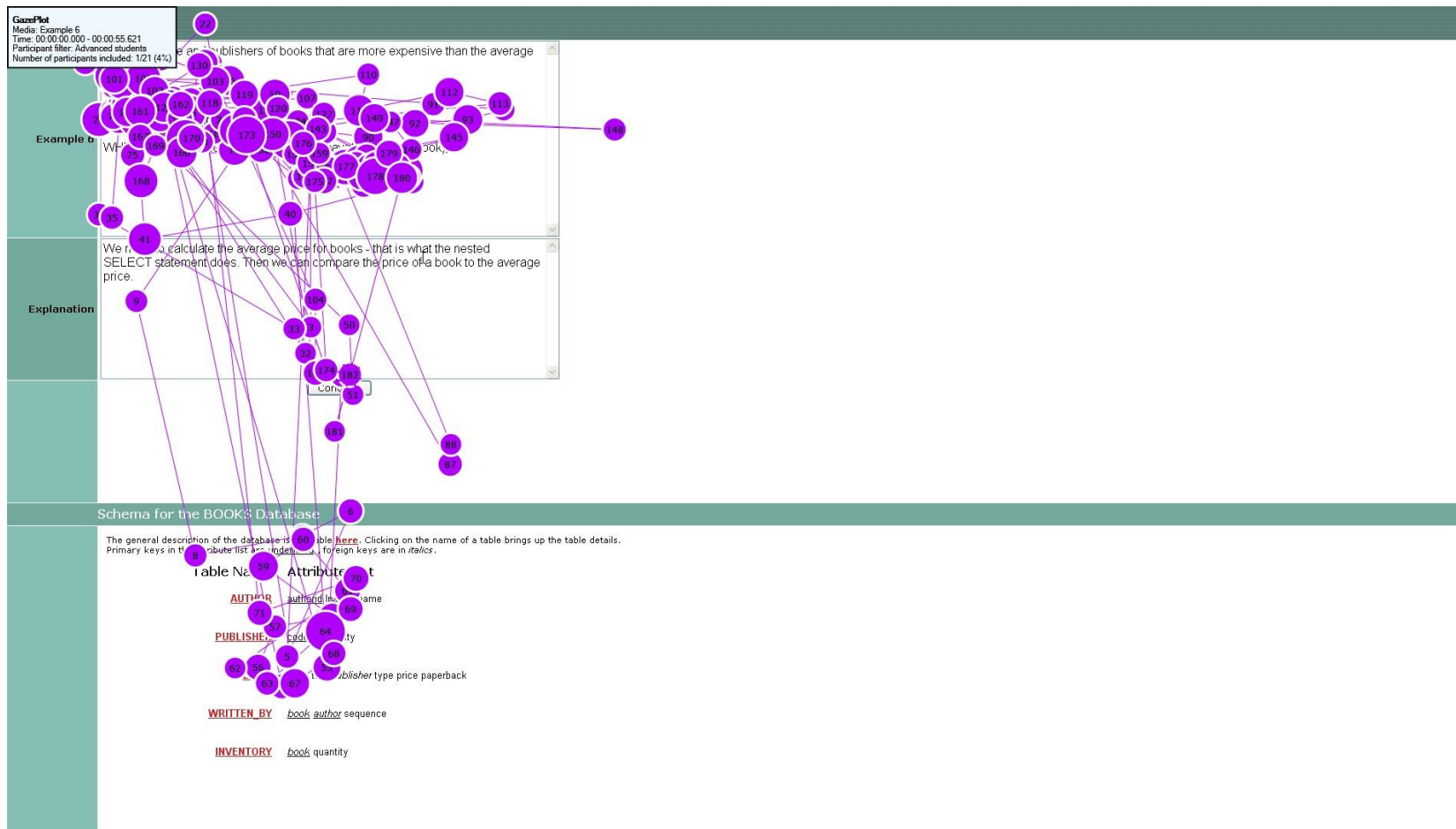


Figure 7.15 Gaze pattern for an advanced student on Example 6

GazePlot
 Media: Example 6
 Time: 00:00:00.000 - 00:00:09.448
 Participant filter: Novices
 Number of participants included: 1/21 (4%)

Example 6

and publishers of books that are more expensive than the average

```

SELECT title, price, publisher, author, title, price
FROM book, publisher
WHERE publisher=code and price>(select avg(price) from book);
    
```

Explanation

We need to calculate the average price for books - that is what the nested SELECT statement does. Then we can compare the price of a book to the average price.

Schema for the BOOKS Database

The general description of the database is available [here](#). Clicking on the name of a table brings up the table details. Primary keys in the attribute list are underlined, foreign keys are in *italics*.

Table Name	Attribute List
AUTHOR	<u>authorid</u> lname fname
PUBLISHER	<u>code</u> name city
BOOK	<u>code</u> title <i>publisher</i> type price paperback
WRITTEN_BY	<i>book</i> <i>author</i> sequence
INVENTORY	<i>book</i> quantity

Figure 7.16 Gaze pattern for a novice student on Example 6

7.4 Discussion

The goal of this study was to analyse how novices and advanced students study SQL examples. Such information enables us to identify the productive and unproductive approaches that students take to study examples. We conducted our study using the example mode of SQL-Tutor and analysed the data from different perspectives.

First, we compared the time novices and advanced students spent studying examples and found there was no significant difference between the two groups. Initially we expected advanced students to spend more time than novices, but eventually the advanced students, who had prior knowledge of the concepts covered in the examples, spent a shorter time studying examples. That is why the standard deviations were very high and the differences were not significant.

The analyses of the students' eye gaze patterns show that advanced students studied the database schema significantly more than novices. We explained that EdE, WeW and EwE are unproductive patterns, and WdW, DE and ED are productive patterns. All novices used the scanning pattern, while only 70% of advanced students have done the same. Behaviour B3 was sometimes used by advanced students, while novices never used this behaviour. B3 indicated that a student started from the worked example area then read the explanation and finally looked at the database schema. Machine learning classifiers also corroborate the patterns analysis and show that advanced students studied the database schema more than novices. We extracted the eye-gaze movement metrics from the eye-tracker software. The result proved that advanced students visited Database AOI more often than novices. Advanced students also spent significantly more time on the database schema than novices.

Overall, the results emphasise the importance of the database schema for advanced students. In our study, students needed database information, such as names and semantics of tables and attributes, to understand examples. Therefore, looking at the database schema was a sign of learning from SQL examples. In contrast, the database schema is not vital to understand the explanations of the examples. The question is, why did novices not pay attention to this crucial area (database schema)? Perhaps novices do not know how to study SQL examples. For instance, they may not know the basic

concepts of primary keys and foreign keys in a database. Therefore, it is interesting to investigate whether or not prompting novices to study a database schema while they study examples would improve students' learning.

It is evident that novices and advanced students use different approaches for studying SQL examples. Such information allows ITSs to be proactive, rather than reactive, to users' actions (Gertner & VanLehn, 2000). Although analysing existing logs of students' interactions with a system can provide some of this information, Bednarik (2005) highlighted the potential for using eye-movement tracking as a source of real-time adaptation.

The presented results suggest new research topics to improve the design of examples. An interesting research question is how to design worked examples to indicate the procedure of studying different AOIs. In this study, we assumed that advanced students follow productive behaviours, but would it be possible to further improve behaviour that advanced students use?

All the analyses performed were based on data captured over the whole session; therefore, the results may change for longer or shorter sessions. It would be interesting to observe how patterns and behaviours change as students become more knowledgeable.

Chapter 8. Conclusion

Prior research shows that students with different prior knowledge benefit differently from studying examples and solving problems. Studies which compared examples with unsupported problem solving show that novices learn more from studying examples than solving unsupported problems. Intelligent Tutoring Systems (ITSs) are different. They provide students with tutored problems. In ITSs, problems can be transformed to examples when students request maximum assistance. If students request a complete solution, the system provides the maximum assistance. While a number of studies show that students learn faster from studying examples than tutored problem solving, there are studies that result in favour of tutored problem solving. Therefore, the results are not yet conclusive.

All prior research compared examples in ITSs about the domains with well-defined tasks; thus, we expanded the research to a domain with ill-defined tasks. We also for the first time investigated the worked-example effect in a constraint-based tutor. We showed that incorporating examples into SQL-Tutor is beneficial; however, with more experience in the domain, students may benefit differently from studying examples. Once students become advanced, examples may cause negative impacts due to the expertise reversal effect. Therefore, we proposed and evaluated an adaptive model that adjusts the number of examples students see to students' prior knowledge. We also investigated students' behaviour while they worked with SQL-Tutor. We found that advanced students paid more attention to the database schema than novices when they studied SQL examples.

8.1 Overview of the project

We conducted a study to investigate using examples only, alternating examples and problems, and problems only in SQL-Tutor. First we hypothesised that students who have worked examples only learn less than the other two groups. Our second hypothesis was that novices will learn the most from a mixture of examples and problems. We found that students who worked with alternating examples and problems, and problems only learnt more than students who had examples only. Therefore, our first hypothesis was confirmed. Moreover, students who worked with a mixture of examples and problems

learnt more conceptual knowledge than students who were in the problems only condition. From novices and advanced students' perspectives, we found that novices learnt the most from alternating examples and problems; thus, our second hypothesis was confirmed: advanced students learnt the same from alternating examples and problems, and solving problems only.

There is no agreement on how much assistance students need while they solve problems, but research confirms that novices need more assistance than advanced students (Sweller, 2006; Clark et al., 2006; van Gog & Rummel, 2010; Sweller et al., 2011). In the first study, we investigated a fixed sequence of examples and problems in SQL-Tutor. A problem with using a fixed sequence of examples and problems was that advanced students had to study examples some aspects of which they might have already known. This might cause expertise reversal effect. Moreover, novices had to solve a problem after studying an example even though they might not have learnt enough from the example; consequently, they might not be able to solve the following problem. This encouraged us to propose an adaptive model to provide students with examples and problems at the right time. We hypothesised that an adaptive model for providing individualised examples will lead to a better learning time and learning gain in comparison with a fixed sequence of examples and ITS. We used a cognitive efficiency score to decide when students need to see examples and when they need to solve problems. A cognitive efficiency score is calculated from a performance score and a cognitive load score in which the cognitive load scores were specified by students. In our model, the performance scores were dependent on the assistance scores. The assistance score indicates how much assistance students received to solve a problem. We proposed a formula to calculate the assistance score from the level of feedback messages students see while solving a problem.

The adaptive strategy determines the type of task (a worked example, a faded example or a problem to be solved) based on how much assistance the student received in the previous problem. In our study, we used the student model to see how much students learnt about each concept, then faded the steps about the concepts students learnt the most in the previous problem. Next, we compared a fixed sequence of alternating worked examples and tutored problem solving (the best condition in the first study) with the proposed adaptive model.

The results showed that students in the adaptive condition learnt significantly more than their peers who were presented with a fixed sequence of worked examples and problem solving; thus, the hypothesis was confirmed. Moreover, novices from the adaptive condition learnt faster than novices from the control group, while the advanced students from the adaptive condition learnt more than peers from the control group. In order to get deeper insights about how students use examples, we used eye-tracking data to compare novices and advanced students. We hypothesised that novices and advanced students study SQL examples differently. The study was performed in the context of SQL-Tutor. We proposed a new technique named EGPA to analyse eye-gaze patterns. In order to comprehend an SQL example, students require information about tables' names and attributes which are available in a database schema. Thus, if students pay attention to the database schema, they understand SQL examples more easily. We analysed students' eye movement data from different perspectives and found that advanced students paid more attention to the database schema than novices; thus, the last hypothesis was confirmed. In future work, it is possible to use the outcomes of this study to provide proactive feedback.

8.2 Significant findings and contributions

Results of the studies conducted in this project led to a number of findings. We found that using alternating examples and problems, and problems only are superior to using examples only in the context of SQL-Tutor. Moreover, for SQL-Tutor novices learn more from alternating examples and problems than problems only or examples only; advanced students learn the same amount from alternating examples and problems, and problems only. We also found that an adaptive strategy for providing examples is superior to using a fixed sequence of examples and problems in the context of SQL-Tutor. The last study showed that novices and advanced students studied SQL examples differently. We found that advanced students paid more attention to the database schema than novices.

The proposed adaptive model outperformed a fixed sequence of using examples and ITSs. We showed that the fixed sequence of examples and problems is superior to using examples only and problems only in the domain of SQL. Therefore, the adaptive model proposed in this thesis is a significant contribution of the study. Students with different prior knowledge need various amounts of assistance and the adaptive model can provide

such different levels of support for novices and advanced students. We showed that novices benefit the most from the AEP condition, in which students saw example-problem pairs. The proposed adaptive model provides example-problem pairs when students need maximum assistance. When students are advanced, they need to solve more problems and practise the knowledge they have. The adaptive model can provide problems only for advanced students.

Some other contributions are as follows. We proposed a new approach to calculating assistance scores. The assistance score was used to measure the student's performance while solving a problem. Finally, we proposed a new approach (EGPA) to extract patterns from eye-tracking data. EGPA proposes a number of definitions for different types of patterns. We used EGPA for manually detecting patterns. It is possible to develop tools for defining patterns considering the EGPA approach. We used EGPA for our research on SQL-Tutor but we believe that this approach can be used in any research involves eye tracking. EGPA focuses on different AOIs that students visit, and AOIs can be defined for any system that involves eye-tracking. EGPA is not an approach to replace machine learning algorithms. In fact, data from EGPA can be used to have a better insight about students using machine learning algorithms; similar to how we used it in our study.

8.3 Limitations

There are a number of limitations of this research, mostly because of questions which lie outside the scope of the research.

Thirty four students volunteered to participate in the first study. As we had three conditions in that study, we had relatively small numbers of participants in each group (two groups with 11 participants and one group with 12 participants). The eye-tracking study also had 22 participants, of which three students were excluded because of samples' invalidity. While these are reasonable numbers of participants, a larger study may help to make stronger conclusions.

The examples in our study were not procedural examples. This means that the procedure was not explained in the examples. However, we used ill-defined tasks in our studies and for most ill-defined tasks, there is no specific procedure for the solution.

For our first study, we designed 20 examples. A few students went over all examples very fast due to illusion of understanding and completed the learning phase. Thus, a study with a larger number of examples may lead to a stronger conclusion.

In this research we could not measure far transfer gain as it was difficult to convince a reasonable number of students to return to the lab and take a delayed post-test voluntarily.

Our adaptive model fades only one or two steps of a solution. Although it allows the examples to gradually transition to a problem, a fading strategy that fades more steps may have more advantages than the current model.

All the studies conducted in this project were in the context of SQL. Thus, another limitation of the project is its singular domain content focus, so it is difficult to determine whether or not the reported results can be generalised.

8.4 Future directions

A number of research avenues are opened as a result of this research. Firstly, we suggested using conceptual-focused self-explanation after problems and procedural-focused self-explanation after examples. The rationale was based on the findings of prior research. Previous research shows that worked examples increase conceptual knowledge more than procedural knowledge, while problem solving produces results in higher acquisition of procedural knowledge (Schwonke et al., 2009; Kim et al., 2007). Even though we conducted our study based on these findings, time constraints prevented us from investigating the difference between effects of conceptual-focused and procedural-focused self-explanation prompts on learning when they are provided after problems or after worked examples. Given more time, we would have liked to investigate the effects of these two new types of self-explanation for learning from problem solving and studying worked examples, but now, this would be an interesting project for future research.

Secondly, we analysed students' eye movement data from different perspectives, and found that advanced students paid more attention to the database schema than novices. In future work, we can use the outcomes of this study to provide proactive feedback. Moreover, the presented results suggest new research topics to improve the design of

examples. An interesting research question is how to design worked examples to imply the procedure of studying different AOIs of the example. In this study, we assumed that advanced students follow productive behaviours, but would it be possible to further improve behaviour that advanced students use?

Thirdly, the presented adaptive strategy determines the type of task based on the student's cognitive efficiency score on the previous task. The performance element of cognitive efficiency is calculated from the assistance score. In future research, performance scores can be calculated more precisely by using assistance scores and self-explanation scores; as a result, we expect that our model would decide more accurately.

Fourthly, in this research we used SQL tasks which are ill-defined. Therefore, there is a need for more research in this area in order to explore the usage and effectiveness of the presented adaptive strategy in well-defined tasks. In our opinion, we will observe the same improvement in learning well-defined tasks by using the proposed adaptive strategy.

Finally, in this project, explanations in the example mode were not adapted to students' knowledge. It would be interesting to observe how adaptive explanations would improve learning from examples. For instance, in Task 2.2 (Appendix A), we can reduce what the learner need to read by removing DISTINCT part from the explanation.

References

- Aleven, V., Koedinger, K. R., & Cross, K. (1999). Tutoring Answer Explanation Fosters Learning with Understanding Understanding. In *Proceedings of the 9th Int. Conf. on Artificial Intelligence in Education* (pp. 199–206).
- 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.
- Aleven, V., McLaren, B. M., Sewall, J., & Koedinger, K. R. (2009). A New Paradigm for Intelligent Tutoring Systems: Example-Tracing Tutors. *Artificial Intelligence in Education*, 19, 105–154.
- Alexe, C., & Gecsei, J. (1996). A learning environment for the surgical intensive care unit. In C. Frasson, G. Gauthier, & A. Lesgold (Eds.), *Proceedings of Int. Conf. on Intelligent Tutoring Systems* (Vol. 1086, pp. 439–447). Springer Berlin Heidelberg.
- Amalathas, S., Mitrovic, A., & Ravan, S. (2012). Decision-making Tutor: providing on-the-job training for oil palm plantation managers. *Research & Practice in Technology Enhanced Learning*, 7(3), 131–152.
- Amershi, S., & Conati, C. (2007). Unsupervised and supervised machine learning in user modeling for intelligent learning environments. In *Proceedings of the 12th Int. Conf. on Intelligent user interfaces* (pp. 72–81). Honolulu, Hawaii, USA: ACM.
- Amershi, S., & Conati, C. (2009). Combining Unsupervised and Supervised Classification to Build User Models for Exploratory. *Educational Data Mining*, 1(1), 18–71.
- Anderson, J. R. (1996). ACT: A simple theory of complex cognition. *American Psychologist*, 51(4), 355–365.
- Anderson, J. R., Corbett, A., Koedinger, K. R., & Pelletier, R. (1995). Cognitive Tutors: Lessons Learned. *Learning Sciences*, 4, pp. 167–207.
- Anderson, J. R., & Reiser, B. J. (1985). The LISP tutor. *Byte*, 10(4), 159–175.
- Anthony, L., Mankoff, J., Mitchell, T., & Gross, M. (2008). *Developing Handwriting-based Intelligent Tutors to Enhance Mathematics Learning*. Doctoral dissertation, Carnegie Mellon University.

- 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.
- Baghaei, N., Mitrovic, A., & Irwin, W. (2007). Supporting collaborative learning and problem-solving in a constraint-based CSCL environment for UML class diagrams. *Computer-Supported Collaborative Learning, 2*(2-3), 159–190.
- Baker, R. S. J., Corbett, A., Koedinger, K. R., Evenson, S., Roll, I., Wagner, A. Z., Naim, M., Raspat, J., Baker, D. J., Beck, J. E. (2006). Adapting to when students game an intelligent tutoring system. In M. Ikeda, K. D. Ashley, & T.-W. Chan (Eds.), *Proceedings of Int. Conf. on Intelligent Tutoring Systems* (Vol. 4053, pp. 392–401). Springer Berlin Heidelberg.
- Baker, R. S., Roll, I., Corbett, A., & Koedinger, K. R. (2005). Do Performance Goals Lead Students to Game the System? In C.-K. Looi, G. McCalla, B. Bredeweg, & J. Breuker (Eds.), *Proceedings of Int. Conf. on Artificial Intelligence in Education* (pp. 57–64). IOS Press.
- Beck, J., & Stern, M. (1996). Applications of AI in Education. *Crossroads, 3*, 11–15.
- Bednarik, R. (2005). Potentials of eye-movement tracking in adaptive systems. In *Proceedings of the 4th Workshop on the Evaluation of Adaptive Systems* (pp. 1–8).
- Billingsley, W., & Robinson, P. (2005). Towards an intelligent online textbook for discrete mathematics. In *Proceedings of Int. Conf. on Active Media Technology* (pp. 291–296).
- Billingsley, W., Robinson, P., Ashdown, M., & Hanson, C. (2004). Intelligent tutoring and supervised problem solving in the browser. In *Proceedings of Int. Conf. on WWW/Internet* (pp. 806–810).
- Bloom, B. S. (1984). The 2 sigma problem: The search for methods of group instruction as effective as one-to-one tutoring. *Educational Researcher, 13*(6), 4–16.
- Bojko, A. (2009). Informative or misleading? heatmaps deconstructed. *Human-Computer Interaction. New Trends, 30–39*.
- Brandl, K. (2005). Are you ready to “Moodle.” *Language Learning & Technology, 9*(2), 16–23.
- Brown, A. L., & Kane, M. J. (1988). Preschool children can learn to transfer: Learning to learn and learning from example. *Cognitive Psychology, 20*(4), 493–523.

- Brusilovsky, P. (2001). Adaptive hypermedia. *User Modeling and User-Adapted Interaction*, 11(1-2), 87–110.
- Bull, S., Cooke, N., & Mabbott, A. (2007). Visual attention in open learner model presentations: An eye-tracking investigation. In C. Conati, K. McCoy, & G. Paliouras (Eds.), *Proceedings of Int. Conf. on User Modeling* (pp. 187–196). Springer.
- Bunt, A., Conati, C., Huggett, M., & Muldner, K. (2001). On improving the effectiveness of open learning environments through tailored support for exploration. In *Proceedings of 10th Int. Conf. on Artificial Intelligence in Education* (pp. 365–376). San Antonio, Texas.
- Carbonell, J. R. (1970). AI in CAI: An artificial-intelligence approach to computer-assisted instruction. *IEEE Transactions on Man-Machine Systems*, 11(4), 190–202.
- Catrambone, R., & Yuasa, M. (2006). Acquisition of procedures: The effects of example elaborations and active learning exercises. *Learning and Instruction*, 16, 139–153.
- Chandler, P., & Sweller, J. (1991). Cognitive load theory and the format of instruction. *Cognition and Instruction*, 8(4), 293–332.
- Charness, N., Reingold, E. M., Pomplun, M., & Stampe, D. M. (2001). The perceptual aspect of skilled performance in chess: Evidence from eye movements. *Memory & Cognition*, 29(8), 1146–1152.
- Chi, M., De Leeuw, N., Chiu, M. H., & LaVancher, C. (1994). Eliciting self-explanations improves understanding. *Cognitive Science*, 18(3), 439–477.
- Chi, M. T. H., Bassok, M., Lewis, M. W., Reimann, P., & Glaser, R. (1989). Self-explanations: How students study and use examples in learning to solve problems. *Cognitive Science*, 13(2), 145–182.
- Chi, M. T. H., & VanLehn, K. A. (1991). The content of physics self-explanations. *Learning Sciences*, 1(1), 69–105.
- Clancey, W. J. (1979). Tutoring rules for guiding a case method dialogue. *Man-Machine Studies*, 11(1), 25–49.
- Clark, R. C., Nguyen, F., & Sweller, J. (2006). *Efficiency in learning: evidence-based guidelines to manage cognitive load*. San Francisco, USA: Pfeiffer.
- Cohen, J. (1988). *Statistical power analysis for the behavioral sciences* (2nd ed.). Hillsdale, NJ: Lawrence Erlbaum Associates.

- Conati, C., & Merten, C. (2007). Eye-tracking for user modeling in exploratory learning environments: An empirical evaluation. *Knowledge-Based Systems*, 20(6), 557–574.
- Conati, C., Merten, C., Muldner, K., & Ternes, D. (2005). Exploring eye tracking to increase bandwidth in user modeling. In L. Ardissono, P. Brna, & A. Mitrovic (Eds.), *Proceedings of 10th Int. Conf. User Modeling* (Vol. 3538, pp. 357–366). Edinburgh, Scotland, UK: Springer Berlin Heidelberg.
- Corbett, A., Kauffman, L., Maclaren, B., Wagner, A., & Jones, E. (2010). A Cognitive Tutor for genetics problem solving: Learning gains and student modeling. *Educational Computing Research*, 42(2), 219–239.
- Corbett, A., MacLaren, B., Wagner, A., Kauffman, L., Mitchell, A., & Baker, R. S. (2013). Differential Impact of Learning Activities Designed to Support Robust Learning in the Genetics Cognitive Tutor. In H. C. Lane, K. Yacef, J. Mostow, & P. Pavlik (Eds.), *Proceedings of Int. Conf. on Artificial Intelligence in Education* (Vol. 7926, pp. 319–328). Memphis, USA: Springer Berlin Heidelberg.
- Corbett, A. T., & Anderson, J. R. (1994). Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction*, 4(4), 253–278.
- Deeks, A. (2000). *Web Based Assignments in Structural Analysis*. Centre for Educational Development, Nanyang Technological University.
- Dillenbourg, P., & Self, J. A. (1992). People power: A human-computer collaborative learning system. In C. Frasson, G. Gauthier, & G. McCalla (Eds.), *Proceedings of Int. Conf. on Intelligent Tutoring Systems* (Vol. 608, pp. 651–660). Montreal, Canada: Springer Berlin Heidelberg.
- Duchowski, A. (2007). *Eye tracking methodology: Theory and practice* (Vol. 373). Springer.
- Duda, R. O., Hart, P. E., & Stork, D. G. (2012). *Pattern classification*. John Wiley & Sons.
- Elmadani, M., Mitrovic, A., & Weerasinghe, A. (2013). Understanding Student Interactions with Tutorial Dialogues in EER-Tutor. In L. H. Wong, C.-C. Liu, T. Hirashima, P. Sumedi, & M. Lukman (Eds.), *Proceedings of 21st Int. Conf. on Computers in Education* (pp. 30–40). Bali: Asia-Pacific Society for Computers in Education.

- Gerjets, P., Scheiter, K., & Catrambone, R. (2004). Designing Instructional Examples to Reduce Intrinsic Cognitive Load: Molar versus Modular Presentation of Solution Procedures. *Instructional Science*, *32*, 33–58.
- Gerjets, P., Scheiter, K., & Catrambone, R. (2006). Can learning from molar and modular worked examples be enhanced by providing instructional explanations and prompting self-explanations? *Learning and Instruction*, *16*(2), 104–121.
- Gertner, A. S., & VanLehn, K. (2000). Andes: A coached problem solving environment for physics. In G. Gauthier, C. Frasson, & K. VanLehn (Eds.), *Proceedings of Int. Conf. on Intelligent Tutoring Systems* (Vol. 1839, pp. 133–142). Montreal, Canada: Springer Berlin Heidelberg.
- Gluck, K. A., Anderson, J. R., & Douglass, S. A. (2000). Broader Bandwidth in Student Modeling: What if ITS Were “Eye” TS? In G. Gauthier, C. Frasson, & K. VanLehn (Eds.), *Proceedings of Int. Conf. on Intelligent Tutoring Systems* (Vol. 1839, pp. 504–513). Montreal, Canada: Springer Berlin Heidelberg.
- Goldberg, J. H., & Helfman, J. I. (2010). Comparing information graphics: a critical look at eye tracking. In *Proceedings of the 3rd BELIV'10 Workshop: Beyond time and errors: novel evaluation methods for Information Visualization* (pp. 71–78).
- Graesser, A. C., Jackson, G. T., Mathews, E. C., Mitchell, H. H., Olney, A., Ventura, M., Chipman, P., Franceschetti, D., Hu, X., Louwerse, M.M., Person, N.K.. (2003). Why/AutoTutor: A test of learning gains from a physics tutor with natural language dialog. In R. Alterman & D. Kirsh (Eds.), *Proceedings of the 25th Annual Conference of the Cognitive Science Society* (pp. 474–479). Cognitive Science Society.
- Greer, J. E., & McCalla, G. (1994). *Student modelling: the key to individualized knowledge-based instruction*. Springer Verlag.
- Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The WEKA data mining software: an update. *ACM SIGKDD Explorations Newsletter*, *11*(1), 10–18.
- Hattie, J. (2009). *Visible Learning: A Synthesis of Over 800 Meta-Analyses Relating to Achievement*. New York: Routledge.
- Hausmann, R., Nokes, T., VanLehn, K., & Gershman, S. (2009). The design of self-explanation prompts: The fit hypothesis. In *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.
- Holland, J., Mitrovic, A., & Martin, B. (2009). J-LATTE: a Constraint-based Tutor for Java. In S. C. Kong, H. Ogata, H. C. Arnseth, C. K. K. Chan, T. Hirashima, F. Klett, J. H. M. Lee, C. C. Liu, C. K. Looi, M. Milrad, A. Mitrovic, K. Nakabayashi, S. L. Wong, S. J. H. Yang (Eds.), *Proceedings of the 17th Int. Conf. on Computers in Education* (pp. 142–146). Hong Kong: Asia-Pacific Society for Computers in Education.
- Holt, P., Dubs, S., Jones, M., & Greer, J. (1994). The state of student modelling. In *Student modelling: The key to individualized knowledge-based instruction* (pp. 3–35). Springer.
- Jarodzka, H., Scheiter, K., Gerjets, P., & van Gog, T. (2010). In the eyes of the beholder: How experts and novices interpret dynamic stimuli. *Learning and Instruction*, 20(2), 146–154.
- Jarodzka, H., van Gog, T., Dorr, M., Scheiter, K., & Gerjets, P. (2013). Learning to see: Guiding students' attention via a Model's eye movements fosters learning. *Learning and Instruction*, 25, 62–70.
- Kalyuga, S. (2007). Expertise Reversal Effect and Its Implications for Learner-Tailored Instruction. *Educational Psychology Review*, 19, 509–539.
- Kalyuga, S. (2009). Instructional designs for the development of transferable knowledge and skills: A cognitive load perspective. *Computers in Human Behavior*, 25(2), 332–338.
- Kalyuga, S., Chandler, P., & Sweller, J. (1998). Levels of Expertise and Instructional Design. *Human Factors and Ergonomics Society*, 40, 1–17.
- Kalyuga, S., Chandler, P., Tuovinen, J., & Sweller, J. (2001). When problem solving is superior to studying worked examples. *Educational Psychology*, 93(3), 579–588.
- Kalyuga, S., & Sweller, J. (2004). Measuring Knowledge to Optimize Cognitive Load Factors During Instruction. *Educational Psychology*, 96(3), 558–568.
- Kalyuga, S., & Sweller, J. (2005). Rapid dynamic assessment of expertise to improve the efficiency of adaptive e-learning. *Educational Technology Research and Development*, 53(3), 83–93.

- Kardan, S., & Conati, C. (2011). A Framework for Capturing Distinguishing User Interaction Behaviors in Novel Interfaces. In M. Pechenizkiy, T. Calders, C. Conati, S. Ventura, C. Romero, & J. C. Stamper (Eds.), *Proceedings of 10th Int. Conf on Educational Data Mining* (pp. 159–168).
- Kardan, S., & Conati, C. (2012). Exploring gaze data for determining user learning with an interactive simulation. In J. Masthoff, B. Mobasher, M. Desmarais, & R. Nkambou (Eds.), *Proceedings of Int. Conf. User Modeling, Adaptation, and Personalization* (pp. 126–138). Montreal, Canada.
- Kardan, S., & Conati, C. (2013). Comparing and Combining Eye Gaze and Interface Actions for Determining User Learning with an Interactive Simulation. In S. Carberry, S. Weibelzahl, A. Micarelli, & G. Semeraro (Eds.), *Proceedings of Int. Conf. User Modeling, Adaptation, and Personalization* (pp. 215–227). Rome, Italy: Springer.
- Kardan, S., FitzGerald, N., & Conati, C. (2012). Eye Movement Data Analysis Toolkit (EMDAT) User Manual. *University of British Columbia*.
- Kasarskis, P., Stehwien, J., Hickox, J., Aretz, A., & Wickens, C. (2001). Comparison of expert and novice scan behaviors during VFR flight. In *Proceedings of the 11th Int. Conf. on Aviation Psychology*.
- Kaufman, L., & Rousseeuw, P. (1987). Clustering by means of Medoids. In Y. Dodge (Ed.), *Statistical Data Analysis Based on the L₁-Norm and Related Methods* (pp. 405–416). North-Holland, Amsterdam.
- Kim, R. S., Weitz, R., Heffernan, N. T., & Krach, N. (2007). Tutored Problem Solving vs. “Pure” Worked Examples. In *Proceedings of the 31st Annual Conference of the Cognitive Science Society*. Austin, TX: Cognitive Science Society (pp. 3121–3126).
- Koedinger, K., & Alevan, V. (2007). Exploring the Assistance Dilemma in Experiments with Cognitive Tutors. *Educational Psychology Review*, 19(3), 239–264.
- Koedinger, K. R., & Anderson, J. R. (1993). Effective use of intelligent software in high school math classrooms. In *Proceedings of Int. Conf. on Artificial Intelligence in Education* (pp. 241–248). Charlottesville, VA: Association for the Advancement of Computing in Education.
- Koedinger, K. R., & Anderson, J. R. (1998). Illustrating principled design: The early evolution of a cognitive tutor for algebra symbolization. *Interactive Learning Environments*, 5(1), 161–179.

- Koedinger, K. R., Anderson, J. R., Hadley, W. H., Mark, M. A., & others. (1997). Intelligent tutoring goes to school in the big city. *Artificial Intelligence in Education*, 8, 30–43.
- Kojima, K., Miwa, K., & Matsui, T. (2013). Supporting Mathematical Problem Posing with a System for Learning Generation Processes through Examples. *Artificial Intelligence in Education*, 22(4), 161–190.
- Kyun, S., Kalyuga, S., & Sweller, J. (2013). The Effect of Worked Examples When Learning to Write Essays in English Literature. *Experimental Education*, 81(3), 385–408.
- Litchfield, D., Ball, L. J., Donovan, T., Manning, D. J., & Crawford, T. (2010). Viewing another person's eye movements improves identification of pulmonary nodules in chest x-ray inspection. *Experimental Psychology: Applied*, 16(3), 251–262.
- Loboda, T. D., & Brusilovsky, P. (2010). User-adaptive explanatory program visualization: evaluation and insights from eye movements. *User Modeling and User-Adapted Interaction*, 20(3), 191–226.
- Martin, B., & Mitrovic, A. (2005). Using learning curves to mine student models. In L. Ardissono, P. Brna, & A. Mitrovic (Eds.), *Proceedings of 10th Int. Conf. User Modeling* (pp. 79–88). Springer.
- Mathews, M. (2012). *A Framework for Multiple Adaptable Pedagogical Strategies in Intelligent Tutoring Systems*. University of Canterbury.
- Mathews, M., & Mitrovic, A. (2009). Does Framing a Problem-Solving Scenario Influence Learning? In S. C. Kong, H. Ogata, H. C. Arnseth, C. K. K. Chan, T. Hirashima, F. Klett, J. H. M. Lee, C. C. Liu, C. K. Looi, M. Milrad, A. Mitrovic, K. Nakabayashi, S. L. Wong, S. J. H. Yang (Eds.), *Proceedings of 17th Int. Conf. on Computers in Education*. Hong Kong: Asia-Pacific Society for Computers in Education.
- Mathews, M., Mitrovic, A., Lin, B., Holland, J., & Churcher, N. (2012). Do your eyes give it away? Using eye tracking data to understand students' attitudes towards open student model representations. In S. Cerri, W. Clancey, G. Papadourakis, & K. Panourgia (Eds.), *Proceedings of 11th Int. Conf. on Intelligent Tutoring Systems* (pp. 422–427). Chania, Greece: Springer Berlin Heidelberg.
- Mayer, R. E. (2002). *Multimedia Learning* (second.). New York, NY, USA: Cambridge University Press.

- 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.), *Proceedings of 15th Int. Conf. on Artificial Intelligence in Education* (pp. 222–229). Springer Berlin / Heidelberg.
- McLaren, B. M., Lim, S.-J., Gagnon, F., Yaron, D., & Koedinger, K. R. (2006). Studying the effects of personalized language and worked examples in the context of a web-based intelligent tutor. In M. Ikeda, K. D. Ashley, & T.-W. Chan (Eds.), *Proceedings of Int. Conf. on Intelligent Tutoring Systems* (pp. 318–328). Taiwan: Springer Berlin Heidelberg.
- McLaren, B. M., Lim, S.-J., & Koedinger, K. R. (2008). When and How Often Should Worked Examples be Given to Students? New Results and a Summary of the Current State of Research. In *Proceedings of Int. Conf. Cognitive Science Society* (pp. 2176–2181). Austin, TX: Cognitive Science Society.
- Menzel, W. (2006). Constraint-based modeling and ambiguity. *Artificial Intelligence in Education*, 16(1), 29–63.
- Mierswa, I., Wurst, M., Klinkenberg, R., Scholz, M., & Euler, T. (2006). YALE: Rapid Prototyping for Complex Data Mining Tasks. In L. Ungar, M. Craven, D. Gunopulos, & T. Eliassi-Rad (Eds.), *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 935–940). New York, NY, USA: ACM.
- Mills, C., & Dalgarno, B. (2007). A conceptual model for game-based intelligent tutoring systems. In R. J. Atkinson, C. McBeath, S. K. A. Soong, & C. Cheers (Eds.), *Proceedings of Australasian Society for Computers in Learning in Tertiary Education* (pp. 692–702). Singapore: Centre for Educational Development, Nanyang Technological University.
- Mitrovic, A. (1998). Experiences in implementing constraint-based modeling in SQL-Tutor. In B. Goettl, H. Half, C. Redfield, & V. Shute (Eds.), *Proceedings of Int. Conf. on Intelligent Tutoring Systems* (pp. 414–423).
- Mitrovic, A. (2002). NORMIT: A web-enabled tutor for database normalization. In *Proceedings of Int. Conf. on Computers in Education* (Vol. 2, pp. 1276–1280). Asia-Pacific Society for Computers in Education.
- Mitrovic, A. (2003). An Intelligent SQL Tutor on the Web. *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., Koedinger, K., & Martin, B. (2003). A Comparative Analysis of Cognitive Tutoring and Constraint-Based Modeling. In P. Brusilovsky, A. Corbett, & F. de Rosis (Eds.), *Proceedings of Int. Conf. User Modeling* (Vol. 2702, p. 147). Springer Berlin / Heidelberg.
- Mitrovic, A., & Martin, B. (2000). Evaluating the effectiveness of feedback in SQL-Tutor. In *Proceedings of International Workshop on Advanced Learning Technologies* (pp. 143–144).
- Mitrovic, A., & Martin, B. (2003). Scaffolding and fading problem selection in SQL-Tutor. In *Proceedings of 11th Int. Conf. on Artificial Intelligence in Education* (pp. 479–481). Sydney, Australia.
- Mitrovic, A., & Martin, B. (2004). Evaluating adaptive problem selection. In P. M. E. Bra & W. Nejdl (Eds.), *Proceedings of Int. Conf. Adaptive Hypermedia and Adaptive Web-Based Systems* (pp. 185–194). Eindhoven, The Netherlands.
- Mitrovic, A., & Martin, B. (2007). Evaluating the Effect of Open Student Models on Self-Assessment. *Artificial Intelligence in Education*, 17, 121–144.
- Mitrovic, A., Martin, B., & Suraweera, P. (2007). Intelligent tutors for all: Constraint-based modeling methodology, systems and authoring. *IEEE Intelligent Systems*, 22, 38–45.
- Mitrovic, A., Mayo, M., Suraweera, P., & Martin, B. (2001). Constraint-Based Tutors: A Success Story. In L. Monostori, J. Vancza, & M. Ali (Eds.), *Proceedings of 14th Int. Conf. on Industrial and engineering applications of artificial intelligence and expert systems: engineering of intelligent systems* (pp. 931–940). Budapest, Hungary: Springer Berlin / Heidelberg.
- Mitrovic, A., McGuigan, N., Martin, B., Suraweera, P., Milik, N., & Holland, J. (2008). Authoring constraint-based tutors in ASPIRE: A case study of a capital investment tutor. In *Proceedings of Int. Conf. on education multimedia, hypermedia and telecommunications* (pp. 4607–4616).
- Mitrovic, A., & Ohlsson, S. (1999). Evaluation of a constraint-based tutor for a database language. *Artificial Intelligence in Education*, 10(3-4), 238–256.
- Mitrovic, A., & Weerasinghe, A. (2009). Revisiting Ill-Definedness and the Consequences for ITSs. In V. D. R. Mizoguchi (Ed.), *Proceedings of 14th Int. Conf.*

- on Artificial Intelligence in Education* (pp. 375–382). Amsterdam, The Netherlands: IOS Press.
- Mitrovic, A., Williamson, C., Bebbington, A., Mathews, M., Suraweera, P., Martin, B., Thomson, D., Holland, J. (2011). Thermo-Tutor: An intelligent tutoring system for thermodynamics. In *Proceedings of IEEE Global Engineering Education Conference* (pp. 378–385).
- Moreno, R. (2006). When worked examples don't work: Is cognitive load theory at an Impasse? *Learning and Instruction, 16*(2), 170–181.
- Muldner, K., & Conati, C. (2010). Scaffolding meta-cognitive skills for effective analogical problem solving via tailored example selection. *Artificial Intelligence in Education, 20*(2), 99–136.
- Munro, A., Johnson, M. C., Pizzini, Q. A., Surmon, D. S., Towne, D. M., & Wogulis, J. L. (1997). Authoring simulation-centered tutors with RIDES. *Artificial Intelligence in Education, 8*(3-4), 284–316.
- Mwangi, W., & Sweller, J. (1998). Learning to solve compare word problems: The effect of example format and generating self-explanations. *Cognition and Instruction, 16*(2), 173–199.
- Novick, L. R. (1988). Analogical transfer, problem similarity, and expertise. *Experimental Psychology: Learning, Memory, and Cognition, 14*(3), 510.
- Novick, L. R., & Holyoak, K. J. (1991). Mathematical problem solving by analogy. *Experimental Psychology: Learning, Memory, and Cognition, 17*(3), 398.
- Ohlsson, S. (1992). Constraint-based Modelling. *Artificial Intelligence in Education, 3*, 429–447.
- Ohlsson, S. (1996). Learning From Performance Errors. *Psychological Review, 103*, No. 2, 241–262.
- Paas, F. G. (1992). Training strategies for attaining transfer of problem-solving skill in statistics: A cognitive-load approach. *Educational Psychology, 84*(4), 429–434.
- Paas, F. G., & van Merriënboer, J. J. G. (1993). The Efficiency of Instructional Conditions: An Approach to Combine Mental Effort and Performance Measures. *Human Factors, 35*, 737–743.
- Rayner, K. (1995). Eye movements and cognitive processes in reading, visual search, and scene perception. *Studies in Visual Information Processing, 6*, 3–22.

- Rayner, K. (1998). Eye movements in reading and information processing: 20 years of research. *Psychological Bulletin*, *124*(3), 372–422.
- Reisslein, J., Atkinson, R. K., Seeling, P., & Reisslein, M. (2006). Encountering the expertise reversal effect with a computer-based environment on electrical circuit analysis. *Learning and Instruction*, *16*(2), 92–103.
- Renkl, A., & Atkinson, R. K. (2003). Structuring the Transition From Example Study to Problem Solving in Cognitive Skill Acquisition: A Cognitive Load Perspective. *Educational Psychologist*, *38*(1), 15–22.
- Rich, E. (1989). *Stereotypes and user modeling*. Springer.
- Ringenberg, M. A., & VanLehn, K. (2006). Scaffolding problem solving with annotated, worked-out examples to promote deep learning. In M. Ikeda, K. Ashley, & T. W. Chan (Eds.), *Proceedings of 8th Int. Conf. on Intelligent Tutoring Systems* (pp. 625–634). Springer Berlin / Heidelberg.
- Roediger, H. L., & Karpicke, J. D. (2006). The power of testing memory: Basic research and implications for educational practice. *Perspectives on Psychological Science*, *1*(3), 181–210.
- Roll, I., Alevan, V., McLaren, B., & Koedinger, K. (2011). Metacognitive Practice Makes Perfect: Improving Students Self-Assessment Skills with an Intelligent Tutoring System. In G. Biswas, S. Bull, J. Kay, & A. Mitrovic (Eds.), *Proceedings of Int. Conf. on Artificial Intelligence in Education* (pp. 288–295).
- 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., Alevan, V., Renkl, A., & Schwonke, R. (2009a). Worked Examples and Tutored Problem Solving: Redundant or Synergistic Forms of Support? *Topics in Cognitive Science*, *1*(1), 203–213.
- Salden, R. J. C. M., Alevan, V., Schwonke, R., & Renkl, A. (2009b). The expertise reversal effect and worked examples in tutored problem solving. *Instructional Science*, *38*(3), 289–307.
- Salden, R., Koedinger, K., Renkl, A., Alevan, V., & McLaren, B. (2010). Accounting for Beneficial Effects of Worked Examples in Tutored Problem Solving. *Educational Psychology Review*, *22*(4), 379–392.
- San Agustin, J., Skovsgaard, H., Mollenbach, E., Barret, M., Tall, M., Hansen, D. W., & Hansen, J. P. (2010). Evaluation of a low-cost open-source gaze tracker. In

- Proceedings of the Symposium on Eye-Tracking Research & Applications* (pp. 77–80). Austin, Texas: ACM.
- Schwonke, R., Renkl, A., Krieg, C., Wittwer, J., Alevén, V., & Salden, R. (2009). The worked-example effect: Not an artefact of lousy control conditions. *Computers in Human Behavior*, *25*, 258–266.
- Schworm, S., & Renkl, A. (2006). Computer-Supported Example-Based Learning: When Instructional Explanations Reduce Self-Explanations. *Computers and Education*, *46*, 426–445.
- Shareghi Najar, A., & Mitrovic, A. (2013a). Do novices and advanced students benefit differently from worked examples and ITS? In L. H. Wong, C.-C. Liu, T. Hirashima, P. Sumedi, & M. Lukman (Eds.), *Proceedings of 21st Int. Conf. on Computers in Education* (pp. 20–29). Indonesia, Bali: Asia-Pacific Society for Computers in Education.
- Shareghi Najar, A., & Mitrovic, A. (2013b). Examples and Tutored Problems: How can Self-Explanation Make a Difference to Learning? In H. C. Lane, K. Yacef, J. Mostow, & P. Pavlik (Eds.), *Proceedings of 16th Int. Conf. on Artificial Intelligence in Education* (Vol. 7926, pp. 339–348). Memphis, USA: Springer, Heidelberg.
- Shareghi Najar, A., Mitrovic, A., & McLaren, B. M. (2014a). Adaptive Support versus Alternating Worked Examples and Tutored Problems: Which Leads to Better Learning? In *22nd Int. Conf. on User Modeling, Adaptation and Personalization* (*in press*). Denmark.
- Shareghi Najar, A., Mitrovic, A., & Neshatian, K. (2014b). Utilizing eye tracking to improve learning from examples. In *16th Int. Conf. on Human-Computer Interaction* (*in press*). Greece.
- Sibert, J. L., Gokturk, M., & Lavine, R. A. (2000). The reading assistant: eye gaze triggered auditory prompting for reading remediation. In *Proceedings of the 13th annual ACM symposium on User interface software and technology* (pp. 101–107). ACM Press.
- Silver, E. A. (1994). On Mathematical Problem Posing. *For the Learning of Mathematics*, *14*(1), 19–28.
- Steichen, B., Schmid, O., Conati, C., & Carenini, G. (2013). Seeing how you're looking- Using Real-Time Eye Gaze Data for User-Adaptive Visualization. In *Proceedings of 21st Int. Conf. on User Modeling, Adaptation and Personalization*. Girona, Spain.

- Suraweera, P., & Mitrovic, A. (2002). KERMIT: a constraint-based tutor for database modeling. In S. A. Cerri, G. Gouarderes, & F. Paraguaçu (Eds.), *Proceedings of 6th Int. Conf. on Intelligent Tutoring Systems* (pp. 377–387). Biarritz, France.
- Suraweera, P., & Mitrovic, A. (2004). An Intelligent Tutoring System for Entity Relationship Modelling. *Artificial Intelligence in Education, 14*, 375–417.
- 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, J., & Cooper, G. A. (1985). The Use of Worked Examples as a Substitute for Problem Solving in Learning Algebra. *Cognition and Instruction, 2*, 59–89.
- Tobii Technology. Tobii TX300 eye tracker product description, 2010, (2010).
- Tobii Technology. Tobii studio 3.0 release notes (2011).
- Tsai, M.-J., Hou, H.-T., Lai, M.-L., Liu, W.-Y., & Yang, F.-Y. (2012). Visual attention for solving multiple-choice science problem: An eye-tracking analysis. *Computers & Education, 58*(1), 375–385.
- Van Gog, T. (2011). Effects of identical example–problem and problem–example pairs on learning. *Computers & Education, 57*(2), 1775–1779.
- Van Gog, T., Jarodzka, H., Scheiter, K., Gerjets, P., & Paas, F. (2009). Attention guidance during example study via the model’s eye movements. *Computers in Human Behavior, 25*(3), 785–791.
- Van Gog, T., & Kester, L. (2012). A Test of the Testing Effect: Acquiring Problem-Solving Skills From Worked Examples. *Cognitive Science*, 1–10.
- 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., & Paas, F. (2008). Instructional Efficiency: Revisiting the Original Construct in Educational Research. *Educational Psychologist, 43*(1), 16–26.
- Van Gog, T., & Rummel, N. (2010). Example-Based Learning: Integrating Cognitive and Social-Cognitive Research Perspectives. *Educational Psychology Review, 22*(2), 155–174.
- Van Merriënboer, J. J. G. (1997). *Training complex cognitive skills: a Four-Component Instructional Design model for technical training* (p. 190). Educational Technology Publications.

- VanLehn, K. (1998). Analogy events: How examples are used during problem solving. *Cognitive Science*, 22(3), 347–388.
- VanLehn, K. (2011). The Relative Effectiveness of Human Tutoring, Intelligent Tutoring Systems, and Other Tutoring Systems. *Educational Psychologist*, 46(4), 197–221.
- VanLehn, K., Lynch, C., Schulze, K., Shapiro, J. A., Shelby, R., Taylor, L., Treacy, D., Weinstein, A., Wintersgill, M. (2005). The Andes Physics Tutoring System: Lessons Learned. *Artificial Intelligence in Education*, 15(3), 147–204.
- Wang, H., Chignell, M., & Ishizuka, M. (2006). Empathic tutoring software agents using real-time eye tracking. In *Proceedings of Int. Conf. on Eye tracking research & applications* (pp. 73–78).
- Webb, N. M. (1989). Peer interaction and learning in small groups. *Educational Research*, 13(1), 21–39.
- Weerasinghe, A., & Mitrovic, A. (2006). Facilitating deep learning through self-explanation in an open-ended domain. *Knowledge-Based and Intelligent Engineering Systems*, 10(1), 3–19.
- Weerasinghe, A., Mitrovic A., van Zijl, M., & Martin, B. (2010). Evaluating the Effectiveness of Adaptive Tutorial Dialogues in Database Design. In *Proceedings of Int. Conf. on Computers in Education* (pp. 33–40). Putrajaya, Malaysia: Asia-Pacific Society for Computers in Education.
- Zakharov, K., Mitrovic, A., & Ohlsson, S. (2005). Feedback micro-engineering in EER-Tutor. In *Proceedings of Int. Conf. on Artificial Intelligence in Education: Supporting Learning through Intelligent and Socially Informed Technology* (pp. 718–725). Amsterdam, The Netherlands.

Appendix A: Examples and explanations

A.1 The material for the first and the second studies (Chapters 5, 6)

Students received the following material either as worked examples or problems to solve. The explanations were shown when students studied worked examples.

Pair 1

1.1. Show the details of all artists.

```
SELECT *  
FROM ARTIST;
```

Explanation:

The SELECT clause allows you to specify what data you want to retrieve from the database. By using * in the SELECT clause you are asking to get all attributes available in tables specified in the FROM clause.

1.2. Show the titles of all songs.

```
SELECT TITLE  
FROM SONG;
```

Explanation:

This query shows all values of the TITLE attribute from the SONG table.

Pair 2

2.1. Show the names of all groups in descending order.

```
SELECT DISTINCT group_name  
FROM in_group  
ORDER BY group_name DESC
```

Explanation:

Some attributes in a table may contain duplicate values. However, sometimes you may want to list only different (distinct) values from a table. The DISTINCT keyword can be used to return only distinct values.

The ORDER BY clause is used to sort the result-set by a specified attribute. The ORDER BY clause sorts the records in ascending order by default (or using ASC). Use the DESC keyword when you want to sort the records in a descending order.

2.2. Show the names of all instruments that artists used, in ascending order.

```
SELECT distinct Instrument  
FROM PERFORMS  
ORDER BY Instrument ASC;
```

Explanation:

In a table, some of the attributes may contain duplicated values. This is not a problem; however, sometimes you may want to list only different (distinct) values from a table. The DISTINCT keyword can be used to return only distinct values.

The ORDER BY keyword is used to sort the result-set by a specified attribute. The ORDER BY keyword sorts the resulting tuples in ascending order by default (or using

ASC). You can use the DESC keyword when you want to sort the records in a descending order.

Pair 3

3.1. Find the CATALOG number of the CD titled “To Record Only Water for Ten Days”.

```
SELECT cat_no
FROM cd
WHERE title='To Record Only Water for Ten Days';
```

Explanation:

The WHERE clause is used to extract those records that fulfil a specified criterion. The query retrieves only those tuples of the CD table where the value of the TITLE attribute is “To Record Only Water for Ten Days”. We used single quotes before and after, because TITLE stores a string.

3.2. Find the first name and the last name of the artist whose ID number is 37.

```
SELECT fname, lname
FROM artist
WHERE id = 37;
```

Explanation:

The WHERE clause is used to extract only those records that fulfil the specified condition.

The query identifies the tuple of the ARTIST table with 37 as the value of the primary key. It then retrieves the first and last name of that artist.

Pair 4

4.1. Show the titles of songs composed by George Gershwin.

```
SELECT title
FROM composer, song_by, song
WHERE song = song.id and composer.id =composer and lname = 'Gershwin' and
fname = 'George';
```

Explanation:

The WHERE clause can contain many conditions, which are used to retrieve only some of the tuples from the given tables or join tables.

If two attributes from two tables have the same name, then we have to use qualified names (table_name.attribute_name).

4.2. Show the surnames of artists in the “Queen” group, as well as the titles of their CDs.

```
SELECT lname, title
FROM artist, in_group, cd
WHERE artist.id= in_group.artist and in_group.group_name='Queen' and
CD.group_name= in_group.group_name;
```

Explanation:

The WHERE clause can contain many conditions, which are used to join tables or select tuples.

If two attributes from two different tables have the same name, then we have to use qualified names (table_name.attribute_name).

Pair 5

5.1. Find the names of artists and instruments they played in “Someone to watch over me” or “Summertime”.

```
SELECT lname , fname, instrument
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');
```

Explanation:

The IN predicate allows you to enumerate values to be used in a comparison to an attribute in the WHERE clause.

5.2. Find the titles of songs and their composers (first name and last name) song 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 song.id=recording.song and 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;
```

Explanation:

The IN predicate allows us to check whether the value of an attribute appears in the enumerated set of values.

Pair 6

6.1. For each group, show the group name and the number of artists.

```
SELECT group_name, count (*)
FROM in_group
GROUP BY group_name;
```

Explanation:

The GROUP BY clause is used to classify the tuples so that all tuples with the same value of group_name are in the same group. There will be as many groups as there are distinct values of the group_name attribute.

COUNT(ARTIST) returns the number of values (NULL values will not be counted) of the ARTIST attribute.

6.2. Show the number of CDs each publisher published.

```
SELECT publisher, count(*)
FROM CD
GROUP BY publisher;
```

Explanation:

The GROUP BY clause is used to group the tuples using the PUBLISHER attribute.

COUNT(*) returns the number of values (NULL values will not be counted) of the TITLE attribute (including duplicates, if there are any).

Pair 7

7.1. Find the IDs of all artists who belong to more than one group. Show the number of groups for each artist.

```
SELECT artist, count(*)
FROM in_group
GROUP BY artist
HAVING count(*)>1;
```

Explanation:

To get the number of groups for each artist, it is necessary to group the tuples first, using the ARTIST attribute first. COUNT(*) returns the number of tuples in each group. The HAVING clause then eliminates those groups of tuples which have a single tuple only.

7.2. Show the number of CDs for each publisher who published more than one CD.

```
SELECT publisher, count (*)
FROM CD
GROUP BY publisher
HAVING count(*)>1;
```

Explanation:

To get the number of CDs per publisher, it is necessary to form groups using the PUBLISHER attribute first. COUNT(title) is applied to each group separately, and returns the number of CDs for each publisher. The HAVING clause then eliminates groups which have a single tuple only.

Pair 8

8.1. For each artist, show his/her id and the number of instruments the artist plays.

```
SELECT artist, count (distinct instrument)
FROM performs
GROUP BY artist;
```

Explanation:

Since we need the required information for each artist, it is necessary to group the tuples so that in each group we have all tuples representing a single artist. Then, we can retrieve the artist ID. To see how many instruments the artist plays, it is necessary to count distinct values of the INSTRUMENT attribute. DISTINCT is necessary as the artist might have played the same instrument in many recordings.

8.2. For each instrument, show how many artists play that instrument.

```
SELECT instrument, count (distinct artist)
FROM performs
GROUP BY instrument;
```

Explanation:

Since we need the required information for each instrument, it is necessary to group the tuples so that in each group we have all tuples representing a single instrument. Then,

we can retrieve the instrument ID. To see how many artists play one instrument, it is necessary to count distinct values of the ARTIST attribute. DISTINCT is necessary as the artist might have played the same instrument in many recordings.

Pair 9

9.1. Show IDs of songs that have more than the average length.

```
SELECT song
FROM recording
WHERE length > (SELECT avg(length)
FROM recording);
```

Explanation:

First we need to calculate the average length of all recordings – that is what the nested SELECT statement does. Then we can compare the length of each recording to the average.

9.2. Find the titles of songs that are shorter than the average length of all recordings.

```
SELECT title
FROM recording join song on recording.song= song.id
WHERE length < (SELECT avg(length)
FROM recording);
```

Explanation:

First we need to calculate the average length of all recordings – that is what the nested SELECT statement does. Then we can compare the length of each recording to the average.

Pair 10

10.1. Find names of artists who recorded every song on the CD titled “The Distance to Here”.

```
SELECT lname, fname
FROM artist
WHERE NOT EXISTS (SELECT id
FROM recording
WHERE NOT EXISTS (SELECT *
FROM contains, cd, performs
WHERE contains.cd=cat_no and
Recording.id=rec and
Performs.artist=artist.id and
Performs.rec= contains.rec and
title='The Distance to Here'));
```

Explanation:

The NOT EXISTS condition is checking whether the nested query returns zero tuples. EXISTS does the opposite (at least one tuple).

10.2. Find the ids of artists who recorded all songs composed by John Davenport.

```
SELECT lname, fname
```

```
FROM artist
WHERE NOT EXISTS (SELECT id
                  FROM recording
                  WHERE NOT EXISTS (SELECT *
                                    FROM song_by, composer, performs
                                    WHERE recording.song=song_by.song
                                    and
                                    Song_by.composer=composer.id and
                                    Performs.rec = recording.id and
                                    Performs.artist=artist.id and
                                    lname='Davenport' and fname='John'));
```

Explanation:

The NOT EXISTS condition is checking whether the nested query returns zero tuples. EXISTS does the opposite (at least one tuple).

A.2 The material for the eye tracking study (Chapter 7)

1. Find titles of books that belong to type science fiction (SFI) or computing (CMP)

```
SELECT title
FROM book
WHERE type='SFI' or type='CMP';
```

Explanation:

The WHERE clause is used to extract tuples that fulfil specified criteria. The query retrieves only those tuples of the BOOK table where the value of the TYPE attribute is “SFI” or “CMP”. We used single quotes before and after, because TYPE stores a string.

2. List details of all authors, ordered by last name.

```
SELECT *
FROM author
ORDER BY lname;
```

Explanation:

The ORDER BY clause is used to sort the resulting tuples by the author’s last name. The ORDER BY clause sorts the tuples in ascending order by default (or using ASC). Use the DESC keyword when you want to sort the records in a descending order.

3. List the different cities that publishers are based in.

```
SELECT distinct city
FROM publisher;
```

Explanation:

Some attributes in a table may contain duplicate values. However, you may want to list only different (distinct) values from a table. The DISTINCT keyword is used to return only distinct values.

4. Produce a list of names of fiction authors ('FIC').

```
SELECT lname, fname
FROM author, book, written_by
WHERE authorid=author and code=book and type='FIC';
```

Explanation:

The WHERE clause can contain many conditions, which are used to join tables or select tuples.

5. For each author, give the author’s name and number of books the author has written. Assign an alias to the total number of books. Order the list in the descending order of author’s last name.

```
SELECT lname, fname, count(*) as BOOKS_WRITTEN
FROM author, written_by
WHERE authorid=author
GROUP BY lname, fname
ORDER BY lname desc;
```

Explanation:

The GROUP BY clause is used to classify the tuples so that all tuples representing books written by the same author are in one group. There will be as many groups as there are authors.

6. List titles, price and publishers of books that are more expensive than the average price.

```
SELECT title, price, publisher.name  
FROM book, publisher  
WHERE publisher=code and price>(select avg(price) from book);
```

Explanation:

We need to calculate the average price for books – that is what the nested SELECT statement does. Then we can compare the price of a book to the average price.

Appendix B: The pre-tests and post-tests

B.1 Pre-Test (first study)

Please answer the following questions. This test is not contributing towards your COSC265 grade.

1. What clause of the SELECT statement allows tuples to be selected?
 - a. SELECT
 - b. FROM
 - c. WHERE
 - d. GROUP BY
 - e. HAVING
 - f. ORDER BY

2. All attributes listed in the ORDER BY clause of a SELECT statement must also appear in the SELECT clause.

True False

3. We need to find the titles of all comedies or dramas. The following SQL statement achieves that.

```
select TITLE  
from MOVIE  
where TYPE = 'comedy' or 'drama';
```

True False

Given the schema below, write solutions for questions 4 and 5.

DEPARTMENT (Dname, Dnumber, Manager, MgrStartDate)

EMPLOYEE (IRD, LName, MInit, FName, BDate, Address, Sex, Salary, Supervisor, DNO)

DEPT_LOCATIONS (DNumber, DLocation)

PROJECT (PName, PNumber, PLocation, DNum)

WORKS_ON (EIRD, PNo, Hours)

DEPENDENT(EIRD, Dependent_Name, Sex, BDate, Relationship)

4. Find the first and last names of all employees who work in the Research department:



5. Find IRDs of all employees whose salary is greater than the average salary of all employees. Show the resulting tuples in descending order.



B.2 Post-Test (first study)

How difficult was it for you to complete the tasks in this study.

Easy					Difficult
1	2	3	4	5	

1. What clause of the SELECT statement allows the resulting table to be sorted?
 - a. SELECT
 - b. FROM
 - c. WHERE
 - d. GROUP BY
 - e. HAVING
 - f. ORDER BY

2. Attribute names used in subqueries are assumed to come from tables used in the outer query.

True False

3. We need to find the titles of all movies that are not comedies. The following SQL statement will achieve that.

```
SELECT TITLE
FROM MOVIE
WHERE TYPE = NOT ('comedy')
```

True False

Questions 4 and 5 are based on the following schema:

DEPARTMENT (Dname, Dnumber, Manager, MgrStartDate)

EMPLOYEE (IRD, LName, MInit, FName, BDate, Address, Sex, Salary, Supervisor, DNO)

DEPT_LOCATIONS (DNumber, DLocation)

PROJECT (PName, PNumber, PLocation, DNum)

WORKS_ON (EIRD, PNo, Hours)

DEPENDENT(EIRD, Dependent_Name, Sex, BDate, Relationship)

4. Find the names of all departments located in Houston.



5. Find IRDs of employees who work on some project for less than the average number of hours for all employees. Show the resulting tuples in ascending order.



B.3 Pre-Test (second study)

1. What clause of the SELECT statement allows tuples to be selected?
 - a. SELECT
 - b. FROM
 - c. WHERE
 - d. GROUP BY
 - e. HAVING
 - f. ORDER BY

2. What clause of the SELECT statement allows conditions to be specified on groups of tuples to produce summary results?
 - a. SELECT
 - b. FROM
 - c. WHERE
 - d. GROUP BY
 - e. HAVING
 - f. ORDER BY

3. What is the effect of the ORDER BY clause?
 - a. Sorts the records in a descending order
 - b. Avoids selecting duplicated records
 - c. Group tuples
 - d. Sorts the record in ascending order

4. Which of the options below is correct?
 - a. COUNT can be used without DISTINCT
 - b. DISTINCT can be specified in ORDER BY
 - c. DISTINCT is always used with COUNT
 - d. DISTINCT is an attribute type

5. All attributes listed in the ORDER BY clause of a SELECT statement must also appear in the SELECT clause.

True False

6. We use IN to define attributes in the WHERE clause.

True False

7. HAVING clause was added to SQL to enhance the readability of the code.

True False

8. Attribute names used in subqueries are assumed to come from tables used in the outer query.

True False

Questions 9 and 10 are based on the following schema:

DEPARTMENT *dname dnumber mgr mgrstartdate*

EMPLOYEE *ird lname minit fname bdate address sex salary supervisor dno*

DEPT_LOCATIONS *dnumber dlocation*

PROJECT *pname pnumber plocation dnum*

WORKS_ON *eird pno hours*

DEPENDENT *eird dependent_name sex bdate relationship*

9. Find the first and last names of all employees who work in the Research department:

10. Find IRDs of all employees whose salary is greater than the average salary of all employees. Show the resulting tuples in descending order.

B.4 Post-Test (second study)

1. What clause of the SELECT statement allows the resulting table to be sorted?
 - a. SELECT
 - b. FROM
 - c. WHERE
 - d. GROUP BY
 - e. HAVING
 - f. ORDER BY

2. What clause of the SELECT statement allows conditions to be specified?
 - a. SELECT
 - b. FROM
 - c. WHERE
 - d. GROUP BY
 - e. HAVING
 - f. ORDER BY

3. What does Distinct do in an SQL query?
 - a. Sorts the records in ascending order
 - b. Returns only different values
 - c. Sorts the result using a specified attribute
 - d. Allows to have duplicated records in a database

4. The COUNT aggregate function counts duplicates if:
 - a. It is used in the GROUP BY clause
 - b. DISTINCT is not used
 - c. DISTINCT is used
 - d. It is used in the WHERE clause

5. We use qualified names for attributes in the WHERE clause to choose tables that are not specified in the FROM.

True False

6. NOT IN allows you to specify a condition on an attribute checking that the value of the attribute does not appear in the enumerated set of values.

True False

7. HAVING clause is applied to each group of tuples.

True False

8. The attributes of tables specified in the outer query are always accessible in the nested query.

True False

Questions 9 and 10 are based on the following schema:

DEPARTMENT *dname dnumber mgr mgrstartdate*

EMPLOYEE *ird lname minit fname bdate address sex salary supervisor dno*


DEPT_LOCATIONS *dnumber dlocation*

PROJECT *pname pnumber plocation dnum*

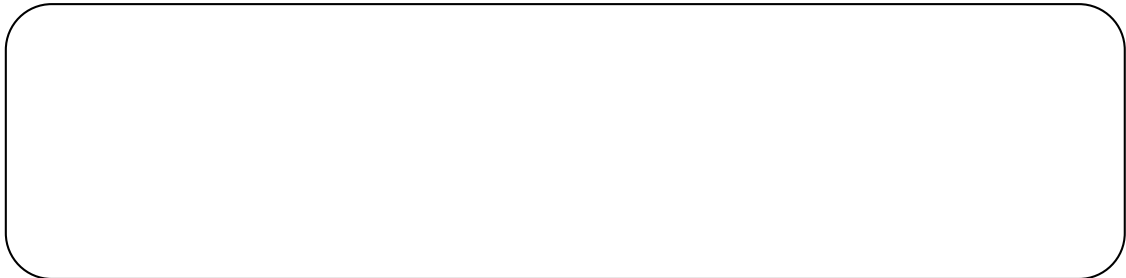
WORKS_ON *eird pno hours*

DEPENDENT *eird dependent_name sex bdate relationship*

9. Select names of all departments located in Houston:

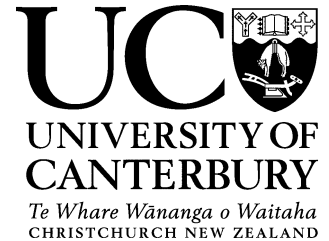


10. Find IRDs of employees who work on some project for less than the average number of hours for all employees. Show the resulting tuples in ascending order.



Appendix C: Information sheet

C.1 Study 1 (presented in Chapter 5)



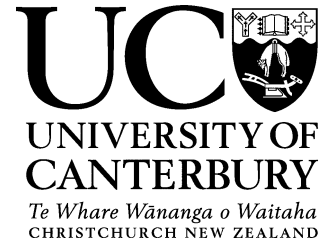
Using examples in ITS Study on SQL-Tutor

Thank you for participating in this evaluation study. The aim of the study is to investigate to which extent the examples and problem-solving can assist you to learn SQL. This study will take roughly 100 minutes. SQL-Tutor may give you only examples, only problems, or a mixture of examples and problems. You are expected to solve the problems provided by the tutor and tutor will provide feedback when your solution is submitted, or review and understand the examples. At the end of each question or example, you will be asked to answer a multiple-choice question. Please, first explain the answer to yourself, and then choose one of the options. At the end of the session, you will be asked to provide feedback about the session.

This project is carried out by Amir Shareghi Najar, who is a PhD student at the Department of Computer Science and Software Engineering, University of Canterbury. He is supervised by Prof. Tanja Mitrovic (tanja.mitrovic@canterbury.ac.nz). He can be contacted through email at (amir.shareghinajar@pg.canterbury.ac.nz). He will be pleased to discuss any concerns you may have about participating in the project. This project has been reviewed and approved by the Dept of Computer Science and Software Engineering and the UCHEC Low Risk Approval process. Complaints may be addressed to The Chair, Educational Research Human Ethics Committee, University of Canterbury, Private Bag 4800, Christchurch, Email: (human-ethics@canterbury.ac.nz).

All the interactions with the SQL-Tutor, pre-test, post-test and the questionnaire responses will be used to analyse whether the example-enriched ITS support has been effective in helping the participants to learn SQL. The data will be analysed by Amir Shareghi Najar and Tanja Mitrovic, and a summary of results will be available to participants. The anonymity of the data will be preserved and they will be kept for about 2 years after the study.

C.2 Study 2 (presented in Chapter 6)



Using examples in SQL-Tutor

Thank you for participating in this evaluation study. The aim of the study is to investigate to which extent the examples and problem-solving can assist you to learn SQL. This test will take roughly 100 minutes. Tutor may give you only examples, only problems, or a mixture of examples and problems. You are expected to solve the problems provided by the tutor and tutor will provide feedback when your solution is submitted, or review and understand the examples. At the end of each question or example, you will be asked to answer a multiple-choice question. Please first explain the answer to yourself, and then choose one of the options. At the end of the session, you will be asked to provide feedback about the session.

This project is carried out by Amir Shareghi Najar, who is a PhD student at the Department of Computer Science and Software Engineering, University of Canterbury. He can be contacted through email at (amir.shareghinajar@pg.canterbury.ac.nz), and will be pleased to discuss any concerns you may have about participating in the study. Amir is supervised by Prof. Tanja Mitrovic (tanja.mitrovic@canterbury.ac.nz). This study has been reviewed and approved by the Department of Computer and Software Engineering, University of Canterbury and the University of Canterbury Human Ethics Committee Low Risk process. Complaints may be addressed to The Chair, Educational Research Human Ethics Committee, University of Canterbury, Private Bag 4800, Christchurch, Email: (human-ethics@canterbury.ac.nz).

All interactions with SQL-Tutor, pre-test, post-test and the questionnaire responses will be used to analyse whether the example-enriched ITS support has been effective in helping the participants to learn SQL. The data will be analysed by Amir Shareghi Najar and Tanja Mitrovic, and a summary of results will be available to participants. The anonymity of the data will be preserved and they will be kept for about 3 years after the study.

The results of the project may be published, but you may be assured of the complete confidentiality of data gathered in this investigation: the identity of participants will not be made public without their consent.

C.3 Study 3 (presented in Chapter 7)



INFORMATION

You are invited to participate as a subject in the study to collect eye-gaze data from students as they interact with the SQL-Tutor intelligent tutoring system.

Your involvement in this project will be to use SQL-Tutor to review some examples and solve a series of problems. You have the right to withdraw from the project at any time without any penalty, including withdrawal of any information provided. Screen recordings will be made during the session.

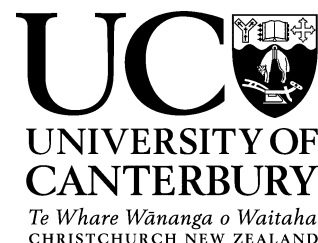
The results of the project may be published, but you may be assured of the complete confidentiality of data gathered in this investigation: the identity of participants will not be made public without their consent. This project is for a PhD thesis, which is a public document that will be made available via the University of Canterbury library database.

The project is being carried out by Amir Shareghi Najar under the supervision of Tanja Mitrovic, who can be contacted at amir.shareghinajar@pg.canterbury.ac.nz and tanja.mitrovic@canterbury.ac.nz respectively. They will be pleased to discuss any concerns you may have about participation in the project.

This proposal has been **reviewed and approved** by the Department of Computer and Software Engineering, University of Canterbury and the University of Canterbury Human Ethics Committee Low Risk process.

Appendix D: Consent form

D.1 Study 1 (presented in Chapter 5)



Consent Form for using examples in ITS study on SQL-Tutor

I have read and understood the description of the above-named project. On this basis, I agree to participate in the project, and I consent on the publication of the results of the project with the understanding that anonymity will be preserved. I understand also that I may at any time withdraw from the project, including withdrawal of any information I have provided.

Signed:

Date:

Name:

D.2 Study 2 (presented in Chapter 6)



Using examples in in SQL-Tutor Consent Form

I have read and understood the description of the above-named study. On this basis, I agree to participate in the study, and I consent on the publication of the results of the study with the understanding that anonymity will be preserved. I understand also that participation is voluntary, and that I may at any time withdraw from the study, including withdrawal of any information I have provided.

Signed:

Date:

Name:

D.3 Study 3 (presented in Chapter 7)



CONSENT FORM

Using Eye Gaze Data to Understand Student Interactions with Examples in Intelligent Tutoring Systems

I have read and understood the description of the above-named project. On this basis I agree to participate as a subject in the project, and I consent to publication of the results of the project with the understanding that anonymity will be preserved and that the resulting thesis will be publicly available via the University of Canterbury library database. In addition, I consent to the screen recordings made during the session to be used in publications, conference presentations and so on provided that my anonymity is preserved.

I understand also that I may at any time withdraw from the project, including withdrawal of any information I have provided. This withdrawal is without any penalty.

I note that this proposal has been reviewed and approved by the Department of Computer and Software Engineering, University of Canterbury and the University of Canterbury Human Ethics Committee Low Risk process.

Name:

User code:

Signature:

Date:

Appendix E: Human ethics committee low risk approval



HUMAN ETHICS COMMITTEE

Secretary, Lynda Griffioen
Email: human-ethics@canterbury.ac.nz

Ref: 2012/03/LR-ERHEC

10 August 2012

Amir Shareghi Najar
Department of Computer Science & Software Engineering
UNIVERSITY OF CANTERBURY

Dear Amir

Thank you for forwarding a copy of your low risk application to the Educational Research Human Ethics Committee for your research proposal titled "Using examples of intelligent tutoring systems".

I am pleased to advise that this application has been reviewed and I confirm support of the Department's approval for this project.

However, the ERHEC wish to recommend that you amend the storage period of data from three to five years.

With best wishes for your project.

Yours sincerely

A handwritten signature in black ink, appearing to read 'N Surtees'.

Nicola Surtees
Chair
Educational Research Human Ethics Committee

"Please note that Ethical Approval and/or Clearance relates only to the ethical elements of the relationship between the researcher, research participants and other stakeholders. The granting of approval or clearance by the Ethical Clearance Committee should not be interpreted as comment on the methodology, legality, value or any other matters relating to this research."

HUMAN ETHICS COMMITTEE

Secretary, Lynda Griffioen
Email: human-ethics@canterbury.ac.nz

Ref: 2013/05/LR-ERHEC

7 August 2013

Amir Shareghi Najar
Department of Computer Science & Software Engineering
UNIVERSITY OF CANTERBURY

Dear Amir

Thank you for forwarding a copy of your low risk application to the Educational Research Human Ethics Committee for your research proposal titled "Using examples in intelligent tutoring systems".

I am pleased to advise that this application has been reviewed and I confirm support of the Department's approval for this project.

With best wishes for your project.

Yours sincerely



Nicola Surtees
Chair
Educational Research Human Ethics Committee

"Please note that Ethical Approval and/or Clearance relates only to the ethical elements of the relationship between the researcher, research participants and other stakeholders. The granting of approval or clearance by the Ethical Clearance Committee should not be interpreted as comment on the methodology, legality, value or any other matters relating to this research."

Appendix F: List of publications

Journal publications:

1. Shareghi Najar, A., & Mitrovic, A. (2014). Examples and tutored problems: is alternating examples and problems the best instructional strategy? *Research and Practice in Technology Enhanced Learning (in press)*.
2. Shareghi Najar, A., Mitrovic, A., & Neshatian, K. (n.d.). Using eye tracking to identify learner differences in example processing. *Artificial Intelligence in Education (submitted)*.
3. Shareghi Najar, A., Mitrovic, A., & McLaren, B. M. (n.d.). Adaptive examples: a learning strategy to incorporate examples into Intelligent Tutoring System. *Computers and Education (submitted)*.

Conference papers:

4. Shareghi Najar, A., Mitrovic, A., & McLaren, B. M. (2014). Adaptive Support versus Alternating Worked Examples and Tutored Problems: Which Leads to Better Learning? In *22nd Int. Conf. on User Modeling, Adaptation and Personalization* (pp. 171–182). Aalborg, Denmark: Springer. **Best student paper award**
5. Shareghi Najar, A., Mitrovic, A., & Neshatian, K. (2014). Utilizing eye tracking to improve learning from examples. In *16th Int. Conf. on Human-Computer Interaction* (pp. 410–418). Greece: Springer.
6. Shareghi Najar, A., & Mitrovic, A. (2013). Do novices and advanced students benefit differently from worked examples and ITS? In L. H. Wong, C.-C. Liu, T. Hirashima, P. Sumedi, & M. Lukman (Eds.), *Proceedings of 21st Int. Conf. on Computers in Education (ICCE)* (pp. 20–29). Indonesia, Bali. **Best student paper award**
7. Shareghi Najar, A., & Mitrovic, A. (2013). Examples and Tutored Problems: How can Self-Explanation Make a Difference to Learning? In K. Yacef et al. (Ed.), *Proceedings of 16th Int. Conf. on Artificial Intelligence in Education (AIED)* (pp. 339–348). Memphis, USA: Springer, Heidelberg.
8. Shareghi Najar, A., & Mitrovic, A. (2012). Should We Use Examples in Intelligent Tutors? In G. Biswas, L.-H. Wong, T. Hirashima, & W. Chen (Eds.), *Proceedings of 20th Int. Conf. ICCE* (pp. 112–114). Singapore: National Institute of Education.

9. Shareghi Najjar, A., & Mitrovic, A. (2012). Using examples in intelligent tutoring systems. In proceedings of Int. Conf. on Intelligent Tutoring Systems (ITS) (pp. 579–581). Chania, Greece.

In addition, we attended two symposiums for computer science students in New Zealand. The details of the presentations are:

10. Shareghi Najjar, A., Mitrovic, A. (2012). Worked-Example effect in Intelligent Tutoring Systems, New Zealand Computer Science Research Student Conference (NZCSRSC), Dunedin, New Zealand.
11. Shareghi Najjar, A., Mitrovic, A. (2013). Towards interleaving examples with problem-solving in SQL-Tutor, New Zealand Computer Science Research Student Conference, (NZCSRSC), Hamilton, New Zealand.

“Publications included in this work have been removed because of copyright restrictions. Please contact the author for more information”.