Department of Computer Science and Software Engineering
University of Canterbury

September 2009

Evaluating the Effectiveness of Multiple Presentations for Open Student Model in EER-Tutor

A thesis submitted in partial fulfilment of the requirements
for the Degree of Master of Science in Computer Science
in the University of Canterbury by

Dandi Duan

Supervisor: Professor Dr. Antonija Mitrovic[1]
Associate supervisor: Dr. Neville Churcher[2]

[1] Department of Computer Science and Software Engineering, University of Canterbury

[2] Department of Computer Science and Software Engineering, University of Canterbury

Acknowledgements

First of all, please allow me to say thank-you to my supervisor Professor Antonija Mitrovic, who has steadily supported me throughout the research period. She has given me countless great ideas, explained whatever things I did not understand, and guided me whenever I was lost. Even she has become busier and busier, she has always been so enthusiastic and patient to me, which has been a great encouragement as I sometimes was not confident enough. Completing this research and writing up the thesis has not been an easy thing for me. Without her efforts I could never reach this point. It is my pleasure to have her involved in one of the most important moments in my life.

I also want to thank my co-supervisor Dr. Neville Churcher, especially for his recommendations about the visualization of the OSM at design phase. Thank-you to Dr. Brent Martin as well, who has been the co-leader of the ICTG and has given me great suggestions at our regular group meetings.

Not only my supervisors, but also my ICTG fellows have shown me their strong supports. I must thank Moffat Mathews and Jay Holland for helping me with so many things, such as setting up the system for evaluation; thank Amali Weerasinghe for sharing her opinions about EER-Tutor and students' self-assessment; thank Puramudi Suraweera and Konstantin Zakharov for their previous contributions to EER-Tutor and its student model. Actually all the ICTG members are superb! I am very proud to be one of them.

Finally I would like to thank my family and friends. Thanks to Mum and Dad, who are far away from me but support me in every possible way. A big thank-you to Alex who looked after me very well at the time I struggled with my study. And thanks to Henry Chu and Xianglin Deng for accompanying with me in the department and giving me suggestions about my research.

Table of Contents

List of Figures

List of Tables

Abstract

As one of the central problems in the area of Intelligent Tutoring Systems (ITSs), student modelling has been widely used to assist in systems' decision making and students' learning. On the one hand, by reasoning about students' knowledge in the instructional domain, a system is able to adapt its pedagogical actions in order to provide a customized learning environment. These actions may include individualized problem-selection, tailored instructions and feedback, as well as updating the presentation of student models. On the other hand, students can reflect on their own learning progress by viewing individual Open Student Models (OSMs) and enhance their meta-cognitive skills by learning from the system's estimation of their knowledge levels. It is believed that making the information in the student model available to students can raise students' awareness of their strengths and weaknesses in the corresponding domain and hence allow them to develop a more effective and efficient way of learning.

An OSM has been developed in EER-Tutor. EER-Tutor is a web-enhanced ITS that supports university students in learning conceptual database modelling. Students design Enhanced Entity-Relationship (EER) diagrams and receive different level of feedback in a problem-solving environment. The pedagogical decisions on feedback generation and problem selection are made according to student models. Previously, student models in EER-Tutor are presented to students on request as skill meters. Skill meters have been proved useful in helping students to improve their meta-cognitive skills. However, as the simplest presentation of a student model, skill meters contain very limited information. Some studies show that an OSM with multiple views is more effective since it supports individual preferences and different educational purposes.

The focus of our work is to develop a variety of presentations for the OSM in EER-Tutor. For this purposes, we have modified the system to include not only skill

meters but also other presentation styles. An evaluation study has been performed after the development. Both subjective and objective results have been collected. This thesis presents the extended EER-Tutor, followed by the analysis of the evaluation study.

CHAPTER 1

Introduction

Learning is the process of acquiring new knowledge and skills. As part of education and personal development, learning occurs in everyone’s life. Over several centuries, educationalists have devoted a great amount of efforts to the research of successful teaching and learning. Today, along with the rapid development of computer science and network technologies, modern pedagogy has been brought to a new level. By adequately exploiting computer software and the Internet, educators are able to deliver a variety of new teaching strategies and instruction styles through convenient modern channels, whereas students are able to obtain knowledge and experience in flexible or even personalized environments. The eventual goal is enabling students to learn more effectively and efficiently.

Educators have been widely aware of the fact that the effectiveness of learning is affected by an array of factors, including students’ background knowledge and experiences, personal abilities and motivations, learning environments, and learning goals set by both instructors and students themselves. Since students’ needs are very diverse, a simple learning environment may fail to cater specific needs of individuals [Holt et al., 1994]. Therefore, many researchers have been working in different areas to support individualization of learning process.

Our aim is to develop a learning environment which provides a set of presentation styles for OSM. Such student models are believed to satisfy individual preferences and various teaching purposes [Bull et al., 2007]. There have been a number of studies in the area of multi-view OSM, but only a few have involved with objective evaluation. In our research, we use both subjective questionnaires and objective tests to evaluate the enhanced OSM and then draw the conclusion of how OSM with multiple presentations helps to improve students' meta-cognitive skills.

1.1 Intelligent Tutoring Systems

While technologies are used to support learning processes, the definition of Technology Enhanced Learning (TEL) is very broad and changes continuously. The dynamic nature of TEL causes the existence of heterogeneous computer-based systems and tools, which are used in a wide variety of contexts. Among numerous existing educational systems, we often see eLearning tools, in which teachers or trainers place learning materials, so that students can study online without taking face-to-face interactions. Although eLearning platforms have become globally popular and some involve with pedagogical elements and adaptive hypermedia, they are still limited to providing instructions over the Internet. The ease they bring into teaching and learning processes does not necessarily guarantee a better learning gain because they cannot provide students with personalized assistance and immediate responses according to students' learning performance. If eLearning tools do not hold the key to successful learning, are there any computer-based systems that simulate human teachers and give feedback in real time? The answer to this question is Intelligent Tutoring Systems (ITSs).

An intelligent tutoring system is an adaptive educational system that provides problem-solving environment and customized feedback. One of the goals of ITSs is to provide learners with individualized instruction comparable to one-to-one human tutoring. One-to-one tutoring has been proved to be more effective than traditional classroom style instruction [Bloom, 1984]. It is the best model to assist students in achieving personal highest performance because human tutors can adapt pedagogical actions at any time on the basis of students' current learning state. However, in real situation, it is often not practical for students to receive sufficient individual support. ITSs, as a potential solution to this problem, have been invented under such circumstances. Although current systems are still far from reaching the goal, research in the area of ITS has been constantly growing, together with the development of other related fields, such as Artificial Intelligence, Education, and Psychology.

1.2 Student Modelling

Student modelling has been widely recognized as one of the central problems in the area of ITSs. In order to make correct pedagogical decisions and deliver customized instructions, a system has to model each student's level of knowledge in the instructional domain.

One of the key issues in successfully modelling a student is how to obtain a complete and accurate model. Ideally, a student model should illustrate the student's knowledge state, preferred learning strategies, area of interests, preferred presentation style, and so on [Mitrovic et al., 2001]. However, the task of building such a model can be too difficult to complete. Attempting to model what a student knows correctly is not enough, but attempting to model what a student knows incorrectly is too complicated since the range of misconception is unlimited. Recall that our goal of ITSs is to build computer-based tutors that achieve the same teaching effects as human tutors, and then the student modelling process of ITSs should be close to how human teachers estimate each student's current knowledge state. Previous research has claimed that human teachers, although use very loose models of their learners, are highly effective in what they do [Holt et al., 1994]. Therefore, even using student models with limited precision, an ITS can still be very useful.

There are different approaches to student modelling. Traditionally, student models are invisible to students. They are used only by the system to perform tailored instructions. Students are not aware of the existence of student models. In recent years, more work has been done on presenting student model to the student, known as Open Student Model (OSM). OSM allows students to view their own models and examine the system estimation of their knowledge levels. When students are aware of their strengths and weaknesses, they may adjust their learning strategies at each stage of the learning process.

According to how much the student is involved into the maintenance of their individual models, student modelling may also be classified as passive, active and interactive modelling. Passive modelling does not need any assistance from students; whereas active modelling may require students to help with the modelling process. In the case of interactive modelling, students play an active role in the development of their own models. The student models are made inspectable. Students may be allowed to change their models or negotiate with the system about the accuracy of the models.

There is also regard on the development of OSMs for peers and instructors [Bull and Hghiem, 2002], as well as cross-course OSMs [Bull et al., 2009], but they are outside the scope of our research. Our work is based on passive open student modelling within a single domain.

1.3 Research Overview

In a one-to-one learning environment, a good human tutor is capable to assess a student’s knowledge level based on his/her performance. The tutor may also discuss it with the student on request. In addition, the tutor may be able to explain the estimation in several ways. Depending on the ability of the student and the learning goal, an appropriate way of explanation should be provided to the student. The student is hence aware of his/her own learning progress and is able to reflect on the learning process. At the end, the student should not only learn the domain knowledge, but also develop better meta-cognitive skills.

If the goal of ITSs is to simulate the teaching strategies applied by human tutors, the way of presenting open student model should be in line with the method human tutors use to explain their view of students' knowledge. That means the open student model should cater for individual preferences.

At the first glance, it seems like the change of OSM does not matter much in terms of the increase in learning gain. Since the pedagogical module is the core of ITS, the student module may be considered no more than an auxiliary component. However, the goal of ITSs should be not only supporting learning about a particular domain, but also teaching students effective and efficient ways of learning. This development of meta-cognitive skills has been considered as an important goal of modern ITSs [Mitrovic and Martin, 2007].

In this research, we have modified EER-Tutor by extending its OSM to include multiple views. When students check their learning progresses, they are provided with a few options. They can choose the view they prefer at any point of time and switch between views freely. We hypothesized that students will be able to gain better understanding of their own knowledge levels by using such a student model. This enhancement of meta-cognition may finally lead to positive effects on learning in the domain.

There are three main phases in this research project:

- Design an presentations that are of different styles and complexities and are suitable for EER domain and our teaching goals.
- Implement each of the presentations and integrate them into the student model to form consistent delivery of the system.
- Evaluate the effectiveness of the extended OSM using pre and post tests as well as questionnaire, and analyze both subjective and objective results.

The contribution of this research is in providing an approach to simulate human-to-human interaction in OSM part of ITSs, where the system gives evaluation or suggestions to students about their performance of certain tasks, and students improve their meta-cognitive skills by learning how to learn. The evaluation study of this research shows students' opinions about the developed system and their preferences on the presentations. It also looks into how students' learning in the domain has been affected. These results will guide future improvement and development in the area.

1.4 Thesis Structure

There are six chapters in this thesis. Chapter presents background information and literature reviews relevant to this research project. The chapter consists of three parts, including an overview of ITSs and Constraint-Based Modelling (CBM), details of EER-Tutor and its student model, and current approaches used for OSMs. A list of presentation styles for OSM is introduced. The advantages and disadvantages of each presentation are also analyzed. In addition, examples of using multiple presentations concurrently in one student model are given and discussed.

From Chapter 3, we start presenting the core work of the research. In Section 1 of Chapter 3, the goals and hypotheses of this research are outlined, along with motivations and supportive reasoning. The second section in Chapter 3 focuses on the design part of the research. Each of the presentations that are added into OSM in EER-Tutor is described in great detail. Reasons for choosing such styles are explained. Chapter 4 continues the theme with how the new views are actually implemented. This evolves with underlying structures and calculations, redesign of interfaces, and system integration. Chapter 5 delivers the evaluation study of this research, including what evaluation methods have been used, how the evaluation has been conducted in reality, what results have been collected, plus a thorough analysis of the results.

Finally, Chapter 6 summarizes the overall work of this research. A conclusion is drawn based on the findings from evaluation. Some suggestions are given towards the improvement of OSM in ITSs. Ideas for future directions of this field are also presented.

CHAPTER 2

Background

In this chapter, we present some related works relevant to our context. We start with looking at the typical architecture of Intelligent Tutoring System, as well as the functions of each component in it. Constraint-Based Modelling is then introduced, which is an approach of overcoming the intractable nature of completely and accurately modelling students' knowledge state. The second section gives details into EER-Tutor, which is the test bed for this research project. The student model part of EER-Tutor is highlighted since it is where we had made modifications. In the last section, a number of existing open student models are listed and discussed. We focus on how these models are presented to students, what are their pros and cons, and how students feel about them. We also address why a student model that includes multiple views are more preferable, followed by examples from previous research.

2.1 Intelligent Tutoring Systems

Psychological research has proved that one-to-one human tutoring is more effective than traditional classroom instruction with a learning improvement of two standard deviations [Bloom, 1984]. This finding has inspired and encouraged so many pioneers of research in ITSs, who have strived to build computer-based tutoring systems that deliver instructions in a way that is similar to one-to-one learning. ITSs are considered intelligent because they provide direct customized feedback to learners without the involvement of human beings. These knowledge-based systems attempt to reproduce the performance of human experts and simulate different aspects of human tutors. The development of such ITSs is an interdisciplinary field that is supported by research in artificial intelligence, computer science, education, and psychology.

ITSs can be used in a variety of instructional domains. Previous research has successfully built a number of ITSs for a broad range of fields. Some well-known examples include Cognitive Tutors [Koedinger and Anderson, 1993] for studying algebra and geometry in high school, Lisp-Tutor [Anderson et al., 1995] for learning Lisp programming language, Andes [Gertner and VanLehn 2000] for solving problems in college-level physics, AutoTutor [Graesser et al., 1999] for helping students learn physics, computer literacy and scientific reasoning, SQL-Tutor [Mitrovic, 1998; Mitrovic et al., 2001] for practicing Structured Query Language (SQL) in database management, and EER-Tutor [Suraweera and Mitrovic, 2004] for learning Enhance Entity-Relationship (EER) modelling in database design. Although these ITSs are developed through different approaches, their underlying structures can be abstracted to a consistent form.

2.1.1 Architecture of ITS

Although the existing ITSs are of different types and domains, their fundamental structures remain relatively stable. As shown in Figure 2.1, a typical ITS contains four major components: a domain module, a pedagogical module, a student modeller, and a user interface [Polson and Richardson, 1988; Freedman, 2000].

Domain Module

Research in expertise systems has shown that any domain module or expert module has to contain an abundance of precise and detailed knowledge derived from people who have many years of experiences in a particular domain [Polson and Richardson, 1988]. Therefore, the function of domain module involves codifying large amount of domain concepts as well as their interrelationship. It contains a description of the domain knowledge that the ITS is teaching.

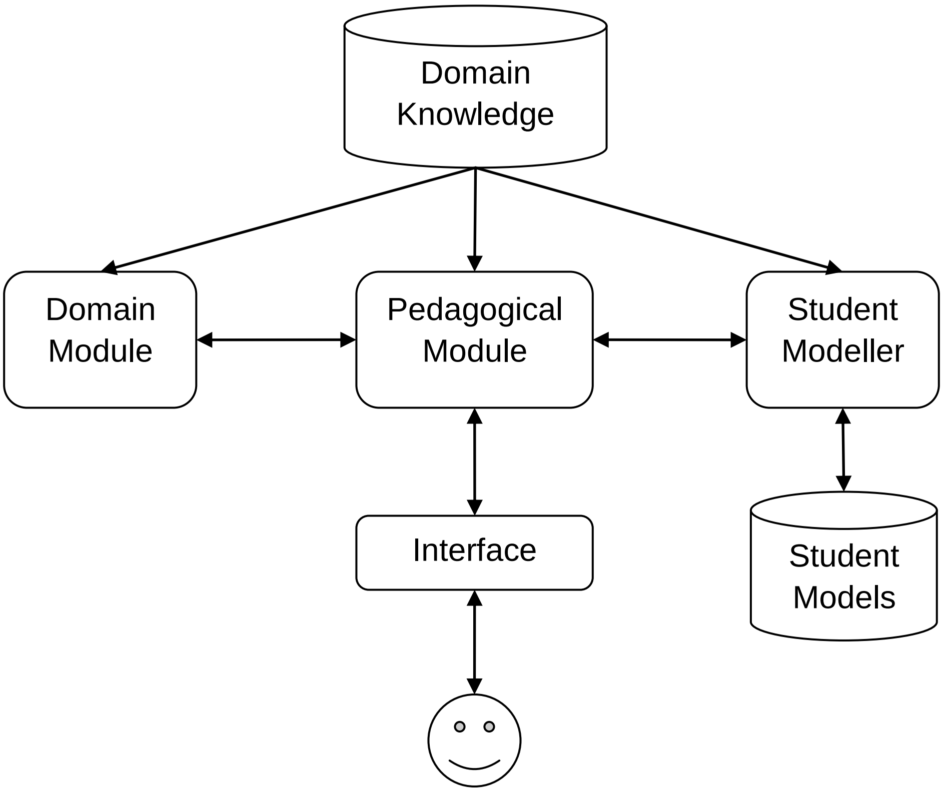


Figure 2.1: Architecture of typical ITS

Student Modeller

The student modeller evaluates students' solutions, captures students' interactions with the system, and uses the data to describe each student's knowledge, including misconceptions and knowledge gaps. The resulting student models are dynamic. They are the system's estimation of each student. The quality of student models is very important because they help to constrain instructional approaches. Pedagogical decisions are made on the basis of information stored in student models.

Pedagogical Module

The pedagogical module works as the core of ITS and interacts with all the other components in the system to form tutoring processes. It receives data from the domain module and student models as inputs, and then takes corrective actions, such as providing feedback, when mismatches between students' knowledge and the domain expert's presumed knowledge are found. These pedagogical actions are finally forwarded to the user interface.

User Interface

The user interface provides the means for students to interact with the ITS. It usually presents domain contents, questions or problems, feedback and solutions, as well as system tools to students.

2.1.2 Constraint-Based Modelling

Constraint-Based Modelling (CBM) was first introduced by Ohlsson as an approach to student modelling [Ohlsson, 1994]. Based on the theory of learning from performance errors [Ohlsson, 1996], CBM was designed to eliminate the necessity of building complete bug libraries and hence conquer the intractability of student modelling. Unlike Model Tracing (MT) approach [Anderson et al., 1995], which traces the sequence of states that a student has gone through in order to solve a problem, CBM concentrates on representing the domain using a set of explicit constraints that are derived from correct solutions. These constraints act like the fundamental principles of the domain. All correct solutions should not violate any of these constraints regardless of problem states; whereas all incorrect solutions are implicitly signified because the violation of constraints indicates errors. In other words, the constraint set partitions the universe of possible solutions into the correct and the incorrect ones [Mitrovic and Ohlsson, 1999].

According to Ohlsson and Rees' introduction of formal notation for constraints, a *state constraint* is an ordered pair <C_r, C_s>, where C_r is the *relevance condition*, and C_s is the *satisfaction condition*. The *relevance condition* determines the class of problem states for which the constraint is relevant. The *satisfaction condition* identifies the class of relevant states in which the constraint is satisfied. Thus, the semantics of a constraint is: *if C_r is true, then C_s must also be true, otherwise something is wrong*.

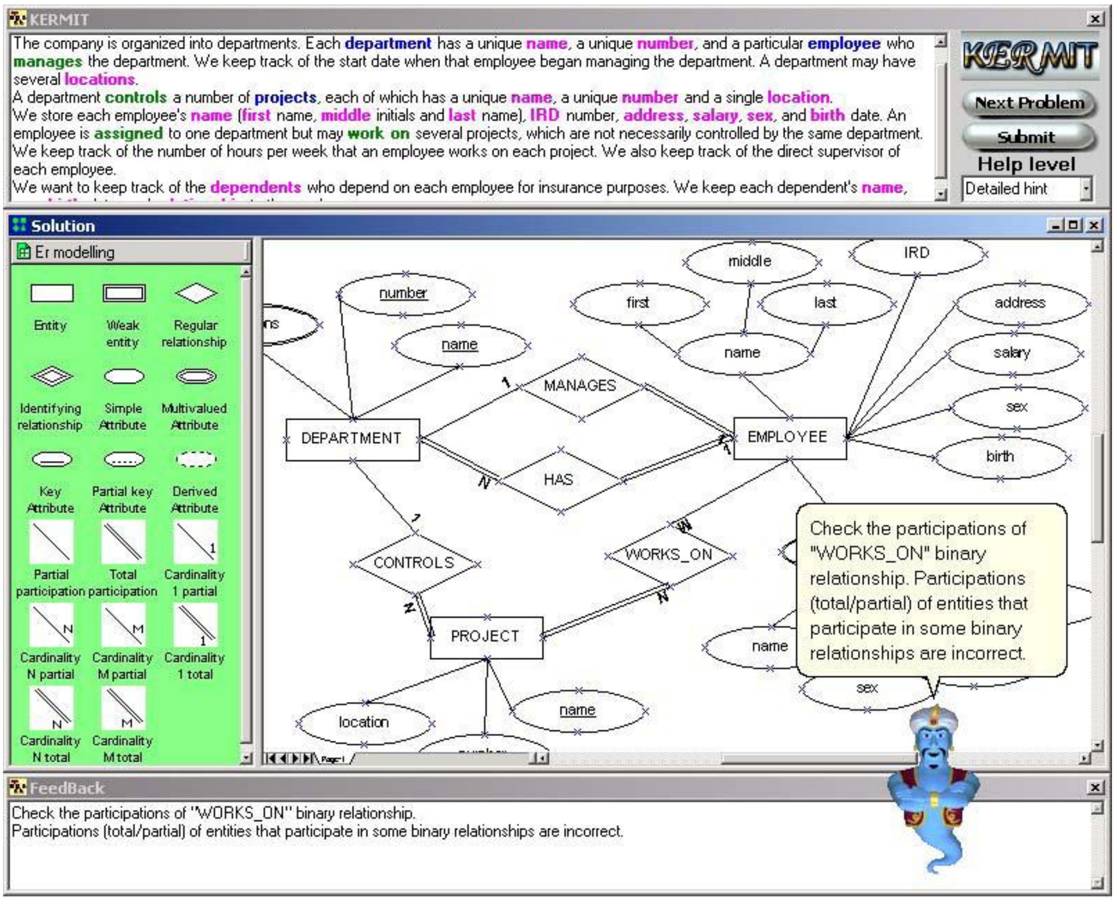
A simple example in daily life is: *if someone is a New Zealand citizen, then he/she must enrol for elections*. In this example, the relevant condition is met when a person is a New Zealand citizen, and then the satisfaction condition has to be met as well in order to result in a correct state, which means, the person must enrol to be registered on the electoral roll.

In terms of student modelling, a student's solution is correct if it satisfies C_s for each constraint whose C_r has been met. Generally, when a student submits a solution to a problem, the solution is matched against C_r in each constraint, resulting in a set of constraints that are relevant to the problem. Each of these constraints is satisfied when its C_s has also been matched. Consequently, the solution is considered correct when all relevant constraints are satisfied. Any violation of C_s will cause error.

It is clear to see that testing whether a known problem state is consistent with a particular set of constraints is trivial. This CBM approach simply reduces the complex computations for modelling students to pattern matching. It does not require a runnable expert module or an extensive bug library, which is difficult to build and takes huge amount of computational time in large domains [Mitrovic and Ohlsson, 1999]. Also, since the sequence of steps that a student arrives at a problem solving process does not matter in CBM, students' solutions are not limited to certain strategies. This characteristic enhances the robustness of constraint-based ITSs, and hence encourages students' creativity.

The Intelligent Computing Tutoring Group (ICTG) at the University of Canterbury has developed a number of constraint-based ITSs for different domains, such as SQL-Tutor for Structured Query Language [Mitrovic, 1998; Mitrovic et al., 2001], EER-Tutor for conceptual database design [Suraweera and Mitrovic, 2002, 2004], NORMIT for data normalization [Mitrovic, 2002, 2005], CAPIT for punctuation and capitalization in English [Mayo and Mitrovic, 2001], and COLLECT-UML for collaborative learning of UML design [Baghaei et al., 2007]. In the following section, we will introduce EER-Tutor in depth since it is the system we used for this research project.

2.2 EER-Tutor


KERMIT

The company is organized into departments. Each **department** has a unique **name**, a unique **number**, and a particular **employee** who **manages** the department. We keep track of the start date when that employee began managing the department. A department may have several **locations**.
A department **controls** a number of **projects**, each of which has a unique **name**, a unique **number** and a single **location**.
We store each employee's **name** (**first** name, **middle** initials and **last** name), **IRD** number, **address**, **salary**, **sex**, and **birth** date. An employee is **assigned** to one department but may **work on** several projects, which are not necessarily controlled by the same department. We keep track of the number of hours per week that an employee works on each project. We also keep track of the direct supervisor of each employee.
We want to keep track of the **dependents** who depend on each employee for insurance purposes. We keep each dependent's **name**,

Help level: Detailed hint

Check the participations of "WORKS_ON" binary relationship. Participations (total/partial) of entities that participate in some binary relationships are incorrect.

FeedBack

Check the participations of "WORKS_ON" binary relationship.
Participations (total/partial) of entities that participate in some binary relationships are incorrect.

Figure 2.2: User interface of KERMIT. Adapted from Suraweera and Mitrovic [2002].

EER-Tutor is a constraint-based ITS in which students can develop skills on Enhanced Entity-Relationship (EER) modelling. Originally, the ICTG has developed KERMIT as a stand-alone constraint-based ITS for teaching basic Entity-Relationship (ER) modelling (shown in Figure 2.2) [Suraweera and Mitrovic, 2002]. Later, KERMIT has been re-implemented to a web-based ITS using WETAS, which is a web-based authoring shell [Martin and Mitrovic, 2002a,b]. KERMIT has been proved to significantly enhance students' performance and has finally been extended to

EER-Tutor [Suraweera and Mitrovic, 2004]. As the successor of KERMIT, EER-Tutor contains a larger set of constraints and problems so that it can handle the enhanced part of database modelling based on EER notations [Zakharov et al., 2005].

2.2.1 Overview of EER-Tutor

As a web-based tutoring system, EER-Tutor is divided into server side and client side. The server processes students' submissions, generates corresponding feedback, and records students' interactions with the system; whereas the client side is presented as a set of dynamic web pages [Zakharov et al., 2005]. The architecture of EER-Tutor is displayed in Figure 2.3.

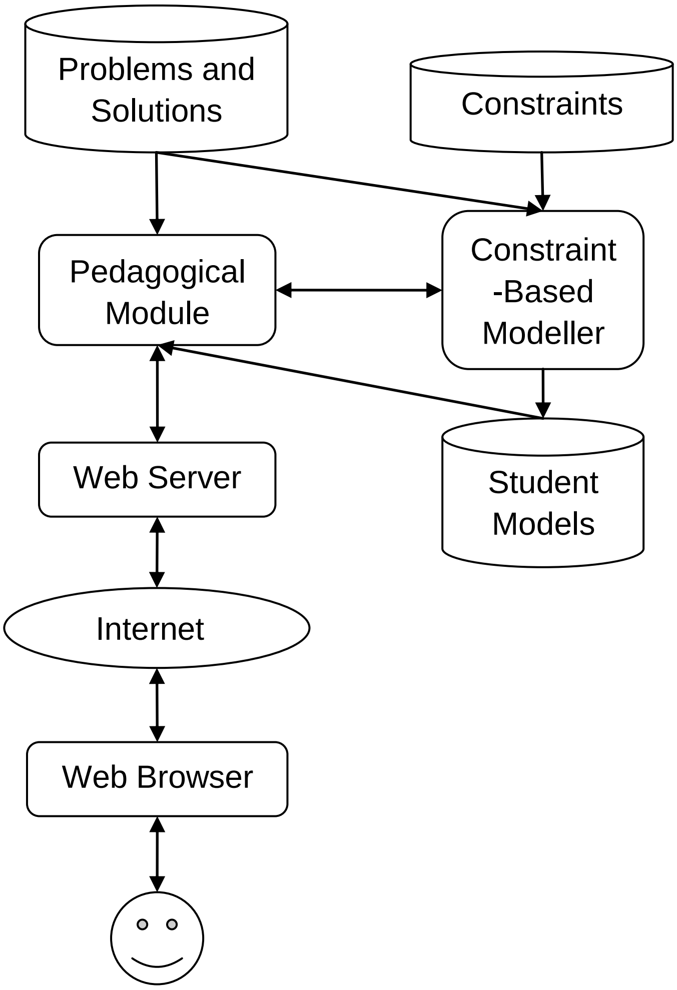
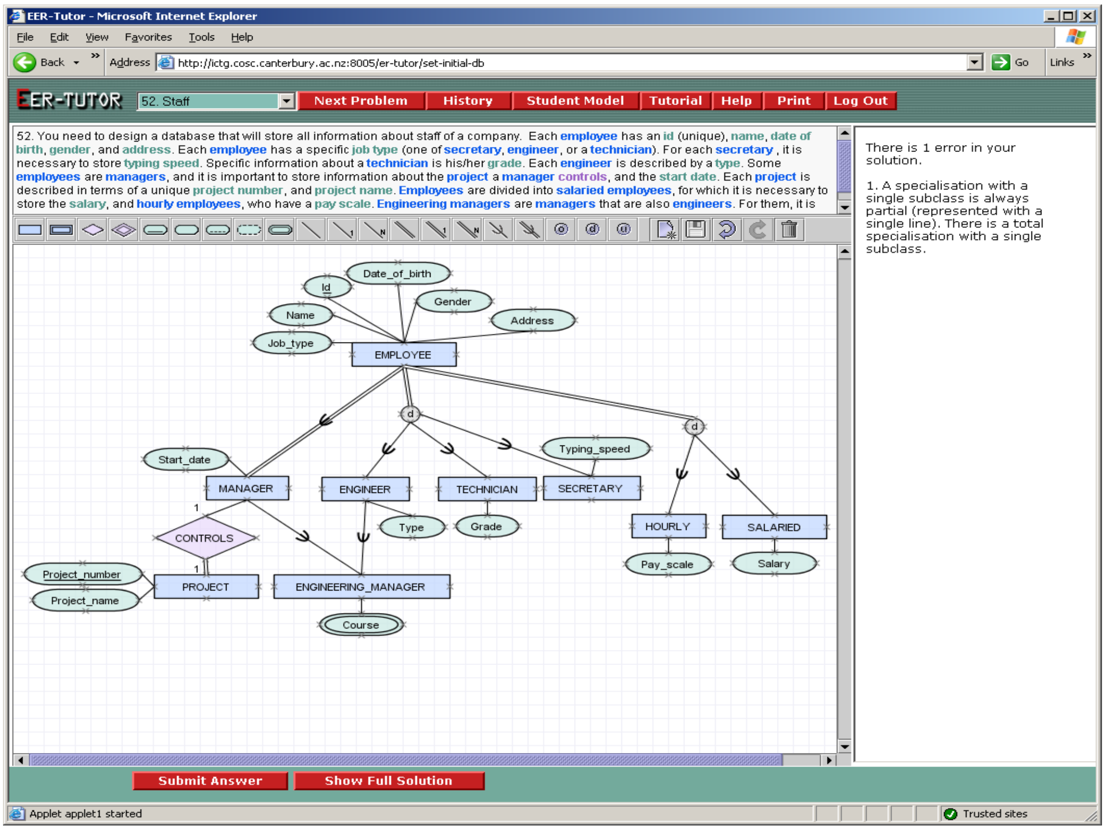


Figure 2.3: Architecture of EER-Tutor.

EER-Tutor provides a problem-solving environment, in which students design EER schemas based on a given set of requirements. Unlike many other ITSs, EER-Tutor does not have a problem solver [Suraweera and Mitrovic, 2002]. Its domain knowledge is represented as a set of syntax and semantic constraints. The database problems and ideal solutions are also predefined. The constraint-based student modeller evaluates each solution by comparing it against the ideal solution according to the system's knowledge base. Student models are maintained and updated dynamically. The pedagogical module drives the entire system by connecting all the components together and making individualized instructional decisions on the basis of student models. Consequently, messages are sent back to the interface to form the exchange of requests and responses between the Web Browser and the Web Server.



52. You need to design a database that will store all information about staff of a company. Each employee has an id (unique), name, date of birth, gender, and address. Each employee has a specific job type (one of secretary, engineer, or a technician). For each secretary , it is necessary to store typing speed. Specific information about a technician is his/her grade. Each engineer is described by a type. Some employees are managers, and it is important to store information about the project a manager controls, and the start date. Each project is described in terms of a unique project number, and project name. Employees are divided into salaried employees, for which it is necessary to store the salary, and hourly employees, who have a pay scale. Engineering managers are managers that are also engineers. For them, it is

There is 1 error in your solution.

1. A specialisation with a single subclass is always partial (represented with a single line). There is a total specialisation with a single subclass.

Figure 2.4: Interface of EER-Tutor. Adapted from Zakharov et al. [2005].

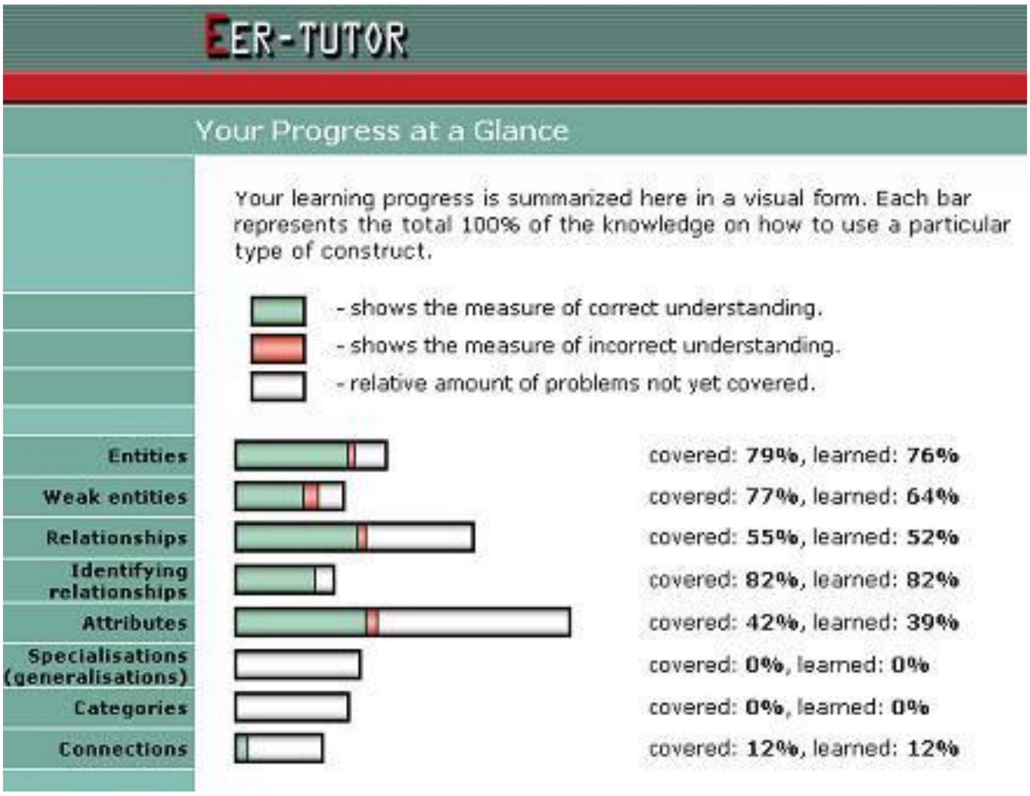
On the client side, EER-Tutor is presented to end-users as HTML pages, which can be displayed using any common Web browser. Figure 2.4 shows the main page of EER-Tutor's interface. On the top of the page, an array of red buttons navigates users to other pages provided by the system, such as student model and tutorial. Beneath that it is the textual description of a database modelling problem. The central area of the page is embedded with a Java applet, which allows students to draw EER models using a given list of tools. Once the student has finished drawing, he/she can click the left button at the bottom to submit the solution. Feedback will be generated immediately and be shown inside the vertical panel on the right. In the case that the student wants to see the ideal solution, he/she can click the other button to receive a pop-up page with full solution.

2.2.2 Student Model in EER-Tutor

Two types of student models are maintained in EER-Tutor: short-term models and long-term models [Suraweera and Mitrovic, 2004]. The short-term student model consists of the satisfaction and violation details of each constraint that is relevant for the current problem state. It is obtained by matching student solutions to ideal solutions and the knowledge base. Every time when a solution is submitted, the student modeller matches it to constraints and determines the ones whose relevance conditions are matched. If a constraint's relevance condition is met, its satisfaction condition is also checked against the problem state. Any violation on the satisfaction condition of a relevant constraint will signal an error in the student solution. This short-term student model is only applicable within the scope of the current problem. The pedagogical module of the system uses the short-term model to generate customized feedback to the student.

In contrast to the short-term student model, the long-term model records the history of each constraint's usage for the student. It contains a list of constraint numbers that the student has encountered. For each of these constraints, there is information on how many times it has been relevant for the student's solutions and whether it has been satisfied or violated each time. This long-term student model is saved into a file every time when the student logs out, and is loaded again when he/she logs in next time. It is used by the pedagogical module to select new problems for the student.

Originally, the long-term student model in EER-Tutor is open to students in the format of skill meters, as shown in Figure 2.5. It is a separate page that can be reached by clicking through the relevant button on the navigation frame of the main page. The page starts with briefly introducing the contents inside student model as a summary of the student's learning progress, followed by short descriptions of colors being used in the meters. There are totally eight progress bars, each of which represents a major concept in the EER domain. The length of a bar indicates the relative amount of knowledge for that particular concept. The green part of a bar specifies how much the student has correctly learned about a concept; the red part gives the measurement of misconception; and the white part shows the relative amount of problems not yet covered. Each bar is also associated with percentages corresponding to covered and learned knowledge. This presentation is a highly abstracted summary of the student model in EER-Tutor.



EER-TUTOR

Your Progress at a Glance

Your learning progress is summarized here in a visual form. Each bar represents the total 100% of the knowledge on how to use a particular type of construct.

🟩 - shows the measure of correct understanding.

🟥 - shows the measure of incorrect understanding.

⬜ - relative amount of problems not yet covered.

Entities	covered: 79%, learned: 76%
Weak entities	covered: 77%, learned: 64%
Relationships	covered: 55%, learned: 52%
Identifying relationships	covered: 82%, learned: 82%
Attributes	covered: 42%, learned: 39%
Specialisations (generalisations)	covered: 0%, learned: 0%
Categories	covered: 0%, learned: 0%
Connections	covered: 12%, learned: 12%

Figure 2.5: OSM in EER-Tutor as skill meters.

2.3 Open Student Model

Student modeller is an essential component for adaptive educational systems. It is responsible for collecting data about all students' activities in the learning environment, including their performance on problem solving and their interactions with the system. By analyzing such data, student modeller is also able to model each student's level of knowledge in the instructional domain. Usually, the resulting student models are able to tell how much each student has learned on each topic, as well as how well each student has performed on each topic. These models are then used to generate customised pedagogical actions, such as problem selection and tailored feedback. Therefore, student modeller is the part that makes a tutoring system adaptive and robust.

As mentioned before, in the case of OSM, the stored information in student model is made available to the corresponding student. Students are able to view the representations of their understanding of each concept in the domain. This is believed to raise student's awareness of their current knowledge levels and encourage them to reflect on the learning processes [Bull, 2004; Bull and Hghiem, 2002; Kay, 1997]. In a system equipped with OSM, students can check their progress over time, know how much and how well they have learned, and adjust their own learning strategies based on their strengths and weaknesses in the domain. By using such a system, they do not only study domain knowledge, but also gradually develop meta-cognitive abilities. They are aware of their own learning states and are capable to carry out self-assessment and self-reflection. The development of these meta-cognitive skills should be taken into account when designing modern ITSs [Mitrovic and Martin, 2007].

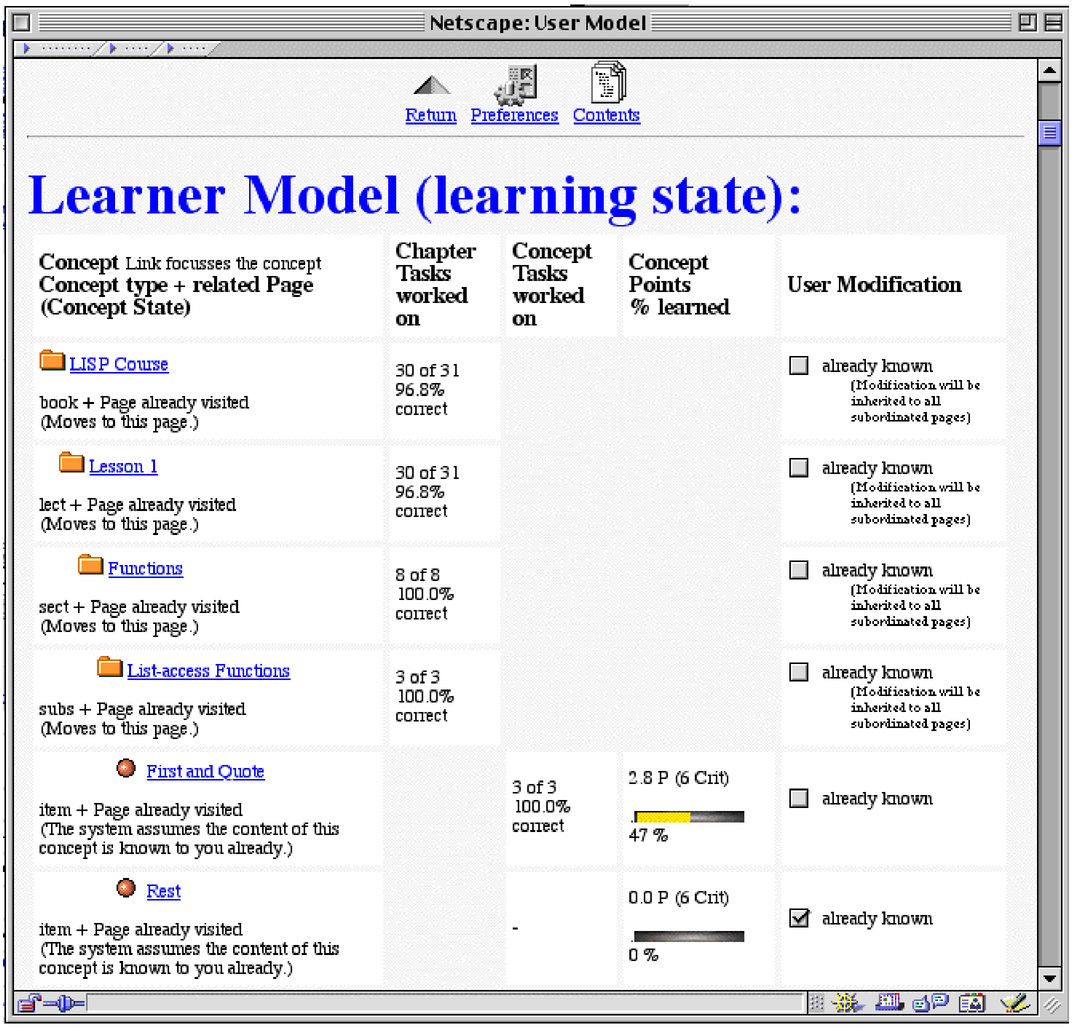
Over the past years, OSM has become increasingly important and interesting. It has been widely developed in educational systems and environments across different domains. Cook and Kay [1997] have used the accessible user model and associated viewer programs for systems that support learning of the *sam* text editor. Weber and Brusilovsky [2001] have opened the user model in ELM-ART, an intelligent interactive educational system for teaching programming in LISP. Dimitrova [2002] has allowed learners to interact with STyLE-OLM, an open learner model demonstrator, in order to generate conceptual representation of their knowledge. Bull and McEvoy [2003] have implemented open learner model in C-POLMILE, an intelligent learning environment for C programming. Zapata-Rivera and Greer [2004] have designed ViSMod, which allows students to create their own models using Bayesian networks. The ICTG has also developed OSM in e-KERMIT [Hartley and Mitrovic, 2002] and SQL-Tutor [Mitrovic, 2003].

Although many educational systems have supported OSMs, the actual representations of student models differ widely. Existing systems have adopted a variety of presentation techniques for OSMs, ranging from simple summaries to complex views. Generally, the simplest views only show high level summaries of students' knowledge on each major topic in the domain, whereas more complicated presentations are likely to display relationships between concepts as well. Moreover, OSM can be either inspection-only or interactive. Ordinary OSM only allows students to view its contents so that they do not have any control over student models. There are also a number of interactive OSMs being designed [Brusilovsky et al., 1996; Dimitrova et al., 2001; Zapata-Rivera and Greer, 2004], which enable students to edit or even negotiate with student models.

Different presentation techniques have shown diverse advantages and disadvantages, therefore they are suitable for different domains and systems. Since we do not implement negotiable student model in this project, we will not discuss related aspects. Rather, we focus on the representation style of each OSM. In the following subsections, we introduce some well-known examples of OSM, including the formats they are using, their pros and cons, and their suitability in terms of system requirements and learning goals.

2.3.1 Skill Meters

Skill meter is one of the most popular formats for OSM. It has been implemented in a number of tutoring systems, such as Geometry Tutor [Aleven and Koedinger, 2000], ELM-ART [Weber and Brusilovsky, 2001], C-POLMILE [Bull and McEvoy, 2003], and SQL-Tutor [Mitrovic and Martin, 2007]. Skill meters are usually displayed as a set of progress bars, each of which indicates the percentage of materials a student has learned for a particular topic or concept [Bull, 2004].

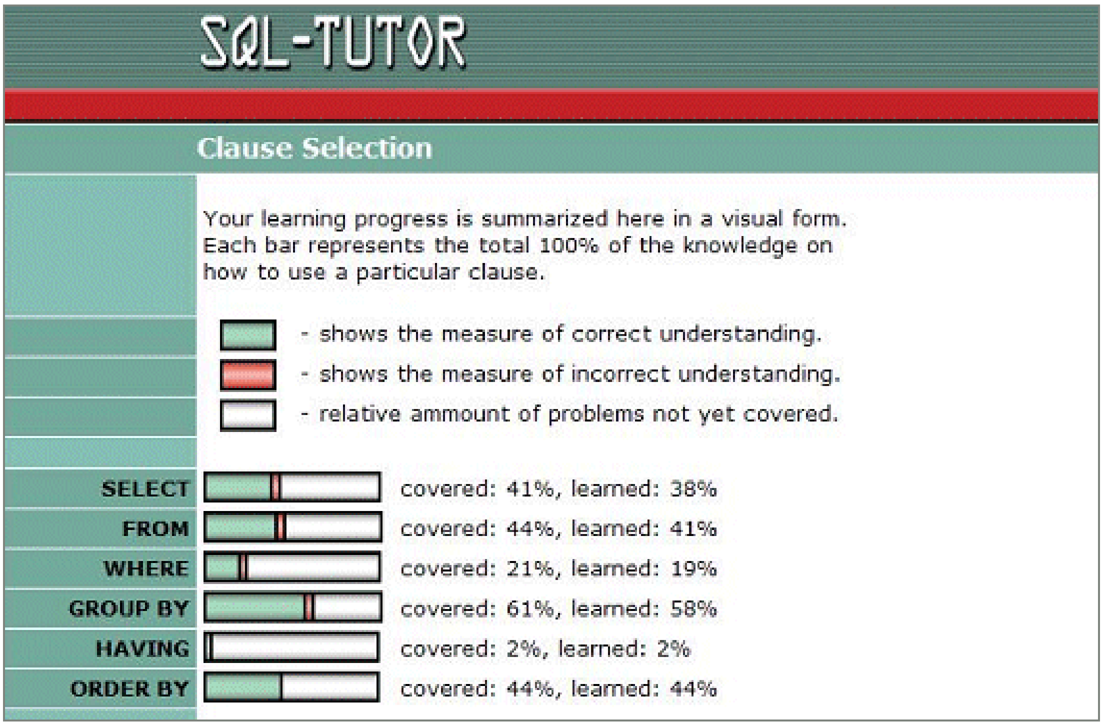


Learner Model (learning state):

Concept Link focusses the concept **Concept type** + **related Page** **(Concept State)**	**Chapter Tasks worked on**	**Concept Tasks worked on**	**Concept Points % learned**	**User Modification**
LISP Course book + Page already visited (Moves to this page.)	30 of 31 96.8% correct			□ already known (Modification will be inherited to all subordinated pages)
Lesson 1 lect + Page already visited (Moves to this page.)	30 of 31 96.8% correct			□ already known (Modification will be inherited to all subordinated pages)
Functions sect + Page already visited (Moves to this page.)	8 of 8 100.0% correct			□ already known (Modification will be inherited to all subordinated pages)
List-access Functions subs + Page already visited (Moves to this page.)	3 of 3 100.0% correct			□ already known (Modification will be inherited to all subordinated pages)
First and Quote item + Page already visited (The system assumes the content of this concept is known to you already.)		3 of 3 100.0% correct	2.8 P (6 Crit) 47 %	□ already known
Rest item + Page already visited (The system assumes the content of this concept is known to you already.)		-	0.0 P (6 Crit) 0 %	☑ already known

Figure 2.6: OSM in ELM-ART. Adapted from Weber and Brusilovsky [2001].

Figure 2.6 shows the OSM in ELM-ART, a web-based ITS designed to teach programming in Lisp. On each page of the course, learners can select a model button to view information about the status of that page. The system displays a table with columns for the page with its current annotation and learning states. The learning states, represented as progress bars, indicate a student's current knowledge of that page [Weber and Brusilovsky, 2001].



SQL-TUTOR

Clause Selection

Your learning progress is summarized here in a visual form. Each bar represents the total 100% of the knowledge on how to use a particular clause.

🟩 - shows the measure of correct understanding.
🟥 - shows the measure of incorrect understanding.
□ - relative ammount of problems not yet covered.

SELECT	covered: 41%, learned: 38%
FROM	covered: 44%, learned: 41%
WHERE	covered: 21%, learned: 19%
GROUP BY	covered: 61%, learned: 58%
HAVING	covered: 2%, learned: 2%
ORDER BY	covered: 44%, learned: 44%

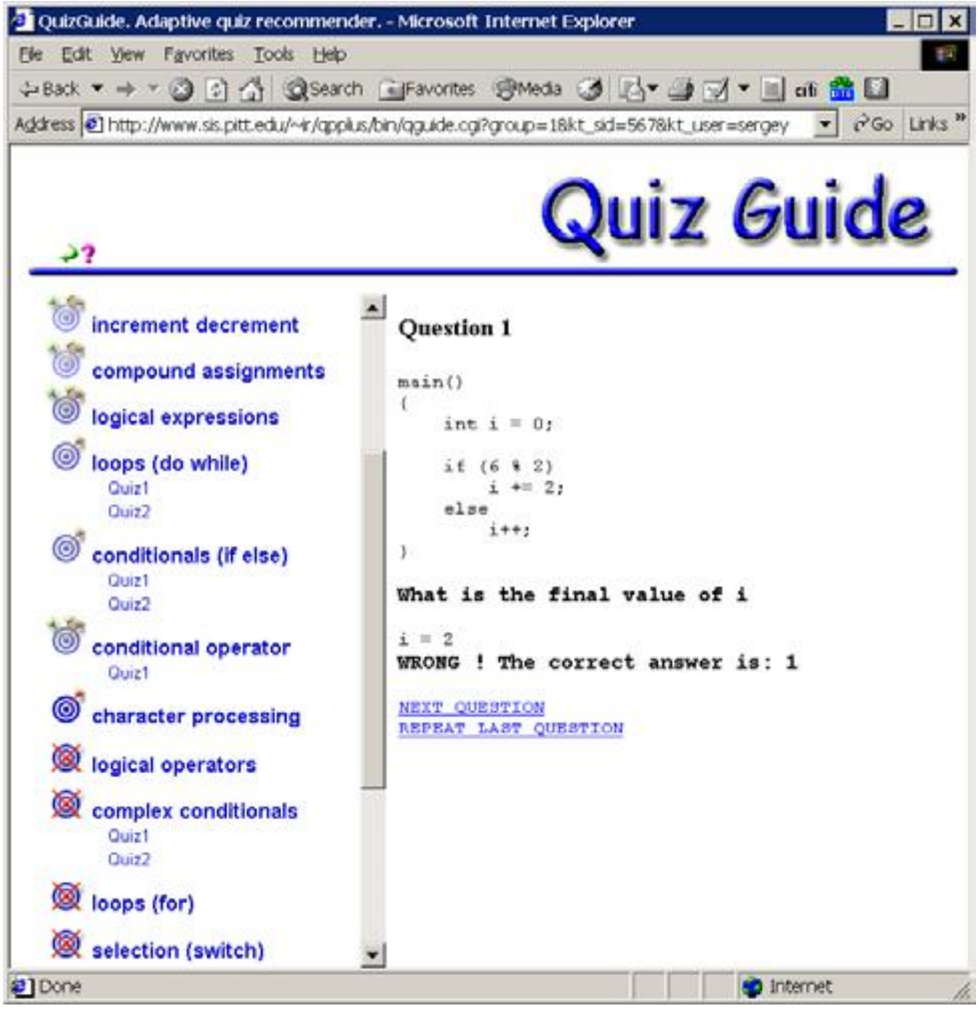
Figure 2.7: OSM in SQL-Tutor. Adapted from Mitrovic and Martin [2007].

In Figure 2.7, we show another example of skill meters. This is the OSM used in SQL-Tutor [Mitrovic and Martin, 2007], an ITS designed by the ICTG to help students learning the database query language SQL. This presentation of OSM is similar to the one used in EER-Tutor, as explained in Section 2.2.

Skill meter has one important advantage of being very intuitive and easy to understand. That means, students do not need to spend much time on "learning" such OSM. However, the simplicity of skill meter may also limit the extent to which OSM can help students to reason about their knowledge. Skill meters are usually high-level summaries of students' learning status in the domain. They do not provide details in terms of students' performance. Also, they may not be able to tell relationships among concepts in the domain.

2.3.2 Target-Arrow Abstraction

Brusilovsky and Sosnovsky [2005] have developed a web-based, self-assessment system, QuizPACK, to provide students with specially constructed classroom quizzes and adaptive guidance. They have also developed the QuizGuide system, which is a value-added service that helps students to select appropriate quizzes. In QuizGuide, they have introduced target-arrow abstraction as a way of presenting OSM.



Question 1

```
main()
{
    int i = 0;

    if (6 % 2)
        i += 2;
    else
        i++;
}
```

What is the final value of i

i = 2
WRONG ! The correct answer is: 1

NEXT QUESTION
REPEAT LAST QUESTION

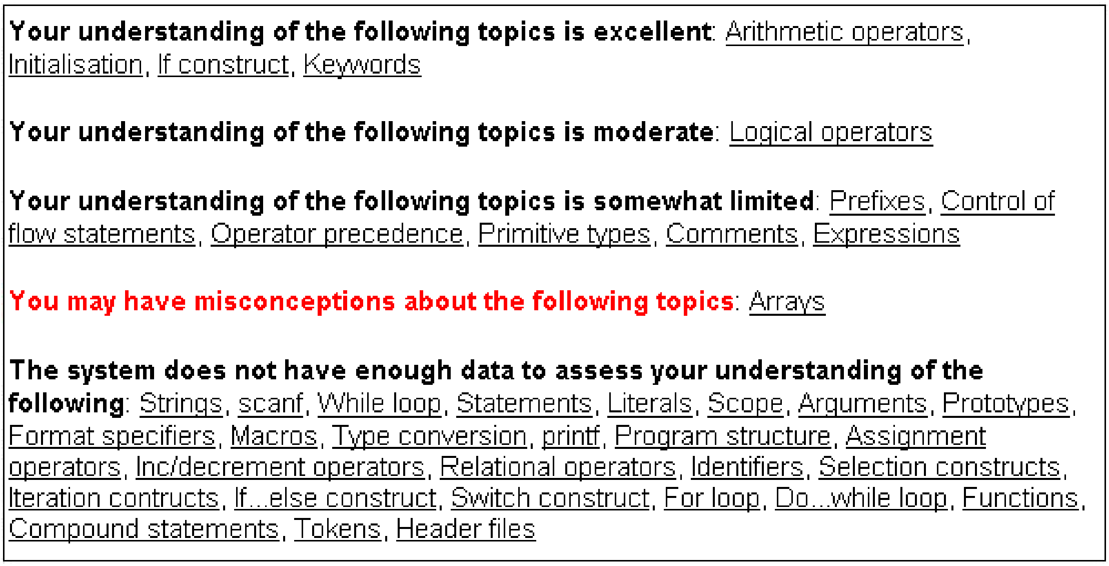
Figure 2.8: OSM in QuizGuide. Adapted from Brusilovsky and Sosnovsky [2005].

As the interface between students and QuizPACK, QuizGuide visualizes the current state of student model by annotating domain topics with target and arrow icons (displayed in Figure 2.8). The color of a target indicates the importance of the corresponding topic as a subgoal of the course. The number of arrows in a target indicates the system estimation of student's knowledge on that topic. Topics that are not ready to be learned are represented as barred targets. Since a student's learning goals and knowledge levels are constantly changing, different views of icons are shown almost every time the student accesses QuizGuide [Brusilovsky and Sosnovsky, 2005].

Same as using skill meter, the use of target and arrow icons exposes a highly abstracted summary of student model. Although students can find out a list of major domain topics and may easily understand what QuizGuide is trying to tell, the system presents only limited information about the underlying models.

2.3.3 Text Summary

Text summary is another simple presentation style for OSM. When using this format, student's domain knowledge is usually described in natural language. Students can read through paragraph-like textual reviews to understand their knowledge levels and find out any misconceptions.



Your understanding of the following topics is excellent: Arithmetic operators, Initialisation, If construct, Keywords

Your understanding of the following topics is moderate: Logical operators

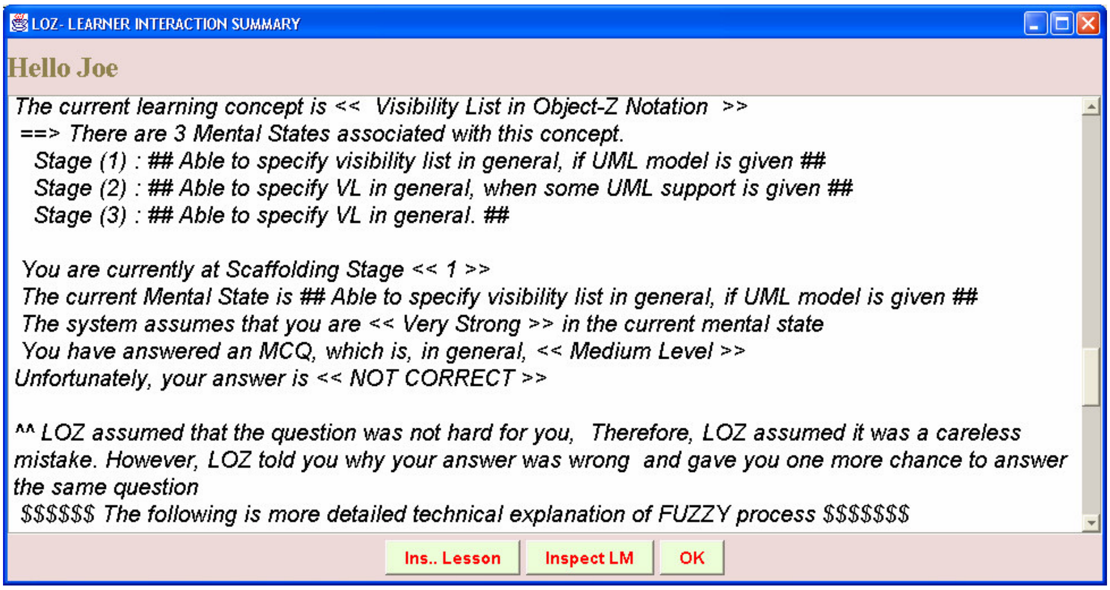
Your understanding of the following topics is somewhat limited: Prefixes, Control of flow statements, Operator precedence, Primitive types, Comments, Expressions

You may have misconceptions about the following topics: Arrays

The system does not have enough data to assess your understanding of the following: Strings, scanf, While loop, Statements, Literals, Scope, Arguments, Prototypes, Format specifiers, Macros, Type conversion, printf, Program structure, Assignment operators, Inc/decrement operators, Relational operators, Identifiers, Selection constructs, Iteration contructs, If...else construct, Switch construct, For loop, Do...while loop, Functions, Compound statements, Tokens, Header files

Figure 2.9: Text summary in Flexi-OLM. Adapted from Bull and Mabbott [2007].

Mabbott and Bull [2007] have included text summary in the open learner model of their system Flexi-OLM, as shown in Figure 2.9. In this view, domain knowledge is represented in a group of concepts, which are categorized into five paragraphs according to student's performance. The first three parts refer to three different levels of student's understanding in domain topics. The fourth part, which is printed in red color, indicates misconceptions occurred in student's learning process. Finally the last part lists all the topics that have not yet been modeled due to the lack of sufficient data.



```
The current learning concept is <<  Visibility List in Object-Z Notation  >>
 ==> There are 3 Mental States associated with this concept.
   Stage (1) : ## Able to specify visibility list in general, if UML model is given ##
   Stage (2) : ## Able to specify VL in general, when some UML support is given ##
   Stage (3) : ## Able to specify VL in general. ##

 You are currently at Scaffolding Stage << 1 >>
 The current Mental State is ## Able to specify visibility list in general, if UML model is given ##
 The system assumes that you are << Very Strong >> in the current mental state
 You have answered an MCQ, which is, in general, << Medium Level >>
Unfortunately, your answer is << NOT CORRECT >>

^^ LOZ assumed that the question was not hard for you,  Therefore, LOZ assumed it was a careless
mistake. However, LOZ told you why your answer was wrong  and gave you one more chance to answer
the same question
 $$$$$$ The following is more detailed technical explanation of FUZZY process $$$$$$$
```

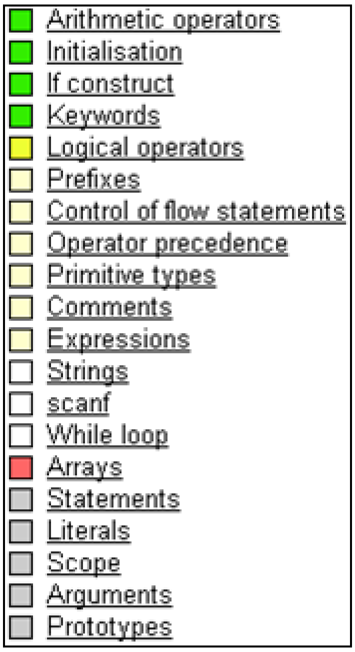
Figure 2.10: OSM in LOZ. Adapted from Mohanarajah et al. [2005].

Another example is a fuzzy logic based learner model used in LOZ, a system for learning Object-Z notation [Mohanarajah et al., 2005]. As displayed in Figure 2.10, since there is fuzzy logic behind the model, fuzzy rule application process can be well described in natural language. Consequently, the view can reflect a human tutor's decision-making process while teaching the student.

According to the existing examples, textual summaries usually imitates human tutors' teaching process sound like human-to-human communication, and hence are easily accepted and understood by students. The special arrangement of paragraphs may be able to focus students' attention onto their level of knowledge. However, the text-only presentation seems less attractive comparing to other views that use graphical visualizations.

2.3.4 Ranked List

Ranked list is another example of views that display concepts according to students' knowledge levels. In a ranked list, domain topics or concepts are ordered by how much the student knows about each of them, usually from high level to low level, and then to misconceptions, with the rest listed as having insufficient data for modelling.



- 🟩 Arithmetic operators
- 🟩 Initialisation
- 🟩 If construct
- 🟩 Keywords
- 🟨 Logical operators
- 🟨 Prefixes
- 🟨 Control of flow statements
- 🟨 Operator precedence
- 🟨 Primitive types
- 🟨 Comments
- 🟨 Expressions
- ⬜ Strings
- ⬜ scanf
- ⬜ While loop
- 🟥 Arrays
- ▨ Statements
- ▨ Literals
- ▨ Scope
- ▨ Arguments
- ▨ Prototypes

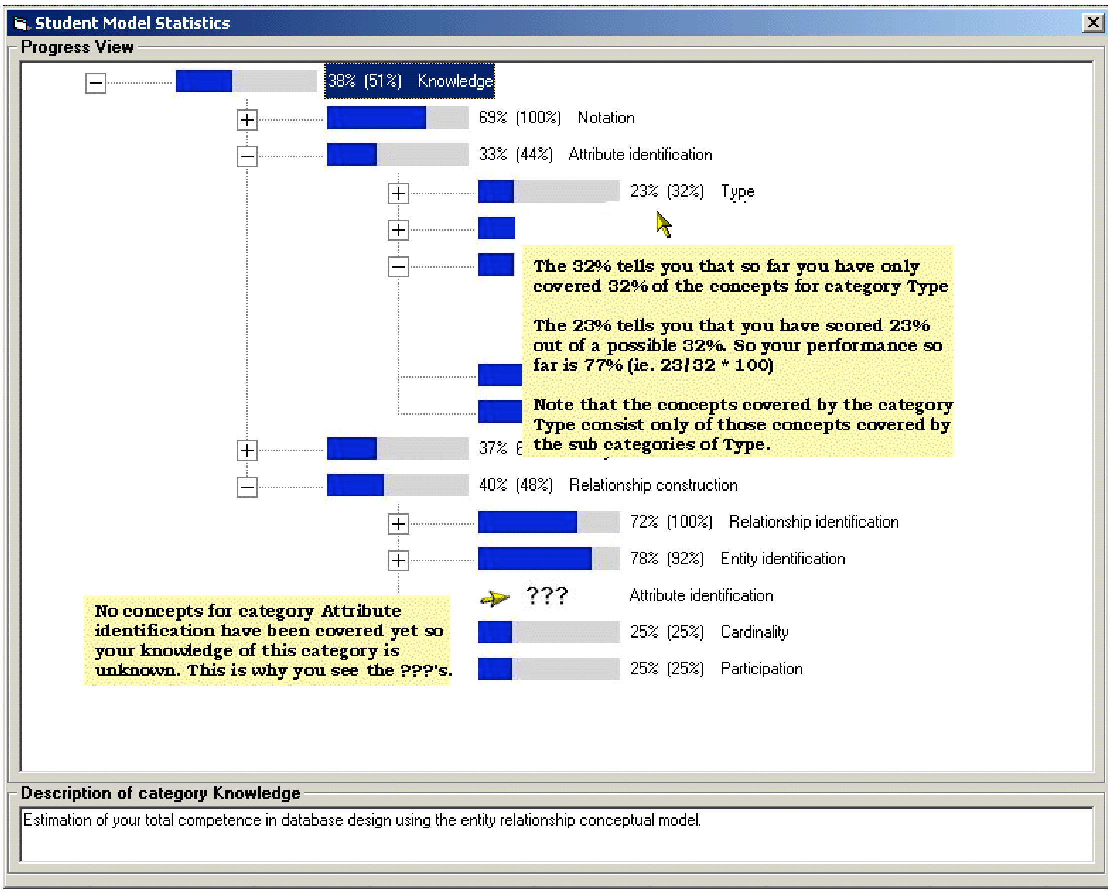
Figure 2.11: Ranked list in Flexi-OLM. Adapted from Bull and Mabbott [2007].

In Flexi-OLM, ranked list is also included. From Figure 2.11, it can be seen that each concept is associated with a colored box. Each color indicates a certain level, such as green for excellent knowledge, yellow for moderate knowledge, white for limited knowledge, and red for misconceptions. At the end, grey boxes are used for unclassified topics [Mabbott and Bull, 2007].

When domain topics in OSM are sequenced according to students' performance, students may easily find out what their strengths and weaknesses are. Also, in a list view, topics may be ranked in other ways and hence deliver other advantages. For example, they can be alphabetically displayed so that users can quickly search certain topics. No matter what logic is used for ranking, there should be enough topics defined in the domain to form a proper list. If there are very few topics, they may be modeled coarsely as know or unknown rather than into different levels, and it may be meaningless to create an alphabetical list.

2.3.5 Concept Hierarchy

All the presentation styles mentioned so far have not described any domain structures. They merely list some or all domain concepts in various ways. In contrast, concept/topic hierarchy is a tree structure built on the basis of conceptual relationships in a domain. As a popular view for OSM, concept hierarchy has been implemented in several systems, such as the *sam* text editor, e-Kermit, and Flexi-OLM.


Student Model Statistics
Progress View
38% (51%) Knowledge
69% (100%) Notation
33% (44%) Attribute identification
23% (32%) Type
37% E
40% (48%) Relationship construction
72% (100%) Relationship identification
78% (92%) Entity identification
???
Attribute identification
25% (25%) Cardinality
25% (25%) Participation
Description of category Knowledge

The 32% tells you that so far you have only covered 32% of the concepts for category Type

The 23% tells you that you have scored 23% out of a possible 32%. So your performance so far is 77% (ie. 23/32 * 100)

Note that the concepts covered by the category Type consist only of those concepts covered by the sub categories of Type.

No concepts for category Attribute identification have been covered yet so your knowledge of this category is unknown. This is why you see the ???'s.

Estimation of your total competence in database design using the entity relationship conceptual model.

Figure 2.12: OSM in e-Kermit. Adapted from [Hartley and Mitrovic, 2002].

In e-Kermit [Hartley and Mitrovic, 2002], a detailed visualization of student model has been developed in the form of a hierarchy (shown in Figure 2.12). All the constraints are grouped pedagogically into domain categories and subcategories; whereas individual constraints are demonstrated as leaves of the tree. Each node in the tree is associated with some progress statistics, which indicate percentages of corresponding materials being taught and correctly learned. Since the concept hierarchy in e-Kermit is designed similar to a directory-view, students are also able to expand or contract certain categories to examine only part of the view in details.

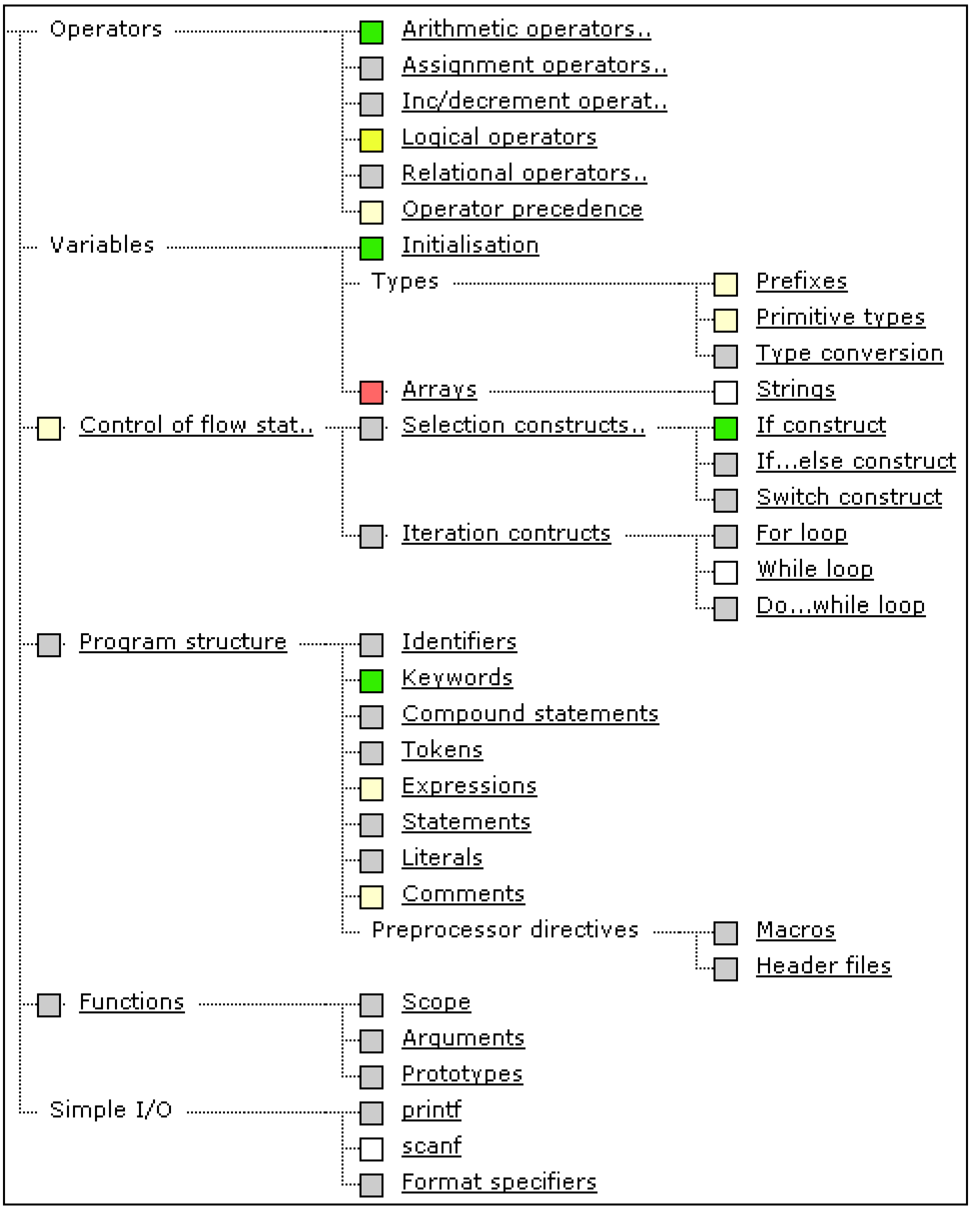


Figure 2.13: Topic hierarchy in Flexi-OLM. Adapted from Bull and Mabbott [2007].

Figure 2.13 shows another example of concept hierarchy. It is the hierarchical structure used in Flexi-OLM to presents all the domain topics. Same as what have been done in the ranked list view, they have included colored boxes as an add-on to the hierarchy in order to indicate different levels of students' knowledge.

Concept hierarchy is different from previous views we have introduced because it does not only display student's learning progress but also demonstrate domain structure. Previous research has found out that students tend to focus less on knowledge levels when using concept hierarchy. Instead, they spend more time on learning the overall structure and the relationships of concepts [Mitrovic and Martin, 2007; Bull and Mabbott, 2007]. Concept hierarchy may include textual description for each concept, and may also use skill meters, progress percentages, or different colors to indicate knowledge levels. It may include only a small number of topics at a time so that students can focus on a portion of the model.

2.3.6 Concept Map

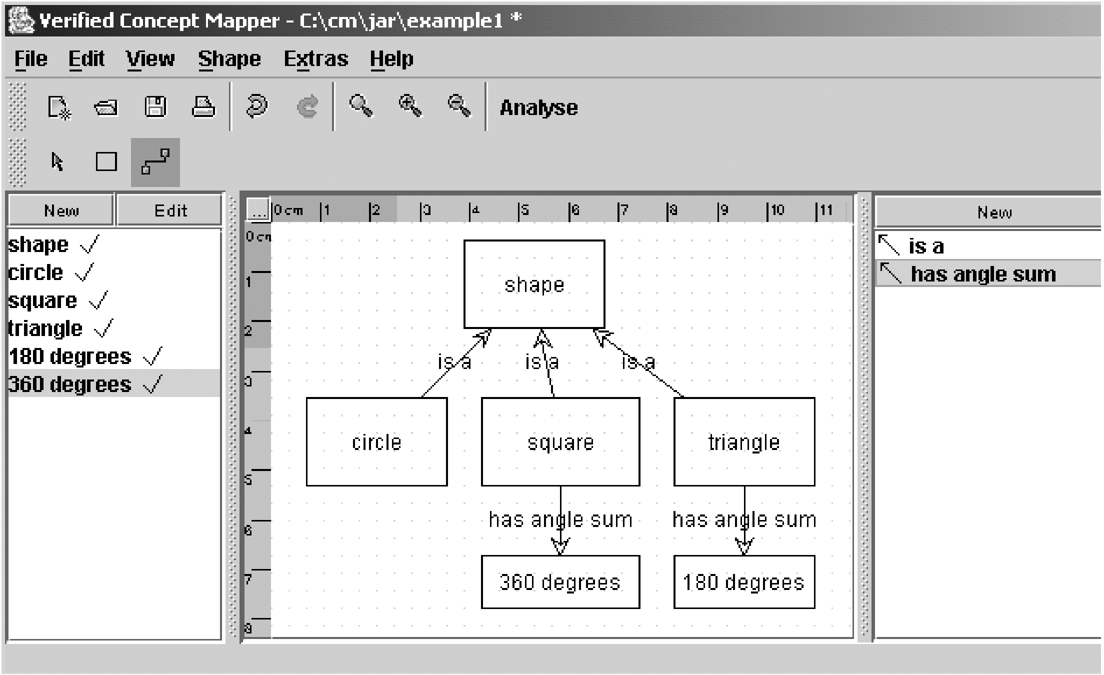


Figure 2.14: Concept map in VCM. Adapted from Cimolino et al. [2003].

Concept map has been widely used for externalizing learner's belief of propositions and relationships in a domain. Current implementations of concept map in student modelling vary in several ways. In some systems, such as STyLE-OLM [Dimitrova, 2003] and VCM [Cimolino et al., 2003] (shown in Figure 2.14), the focus is on leaner constructed maps. In DynMap [Rueda et al., 2006] (shown in Figure 2.15) and COMOV [Perez-Martin and Alfonseca, 2007], it is the system who generates dynamic maps according to student data. Flexi-OLM [Bull and Mabbott, 2006] has also adopted this format, but its concept map is pre-constructed.

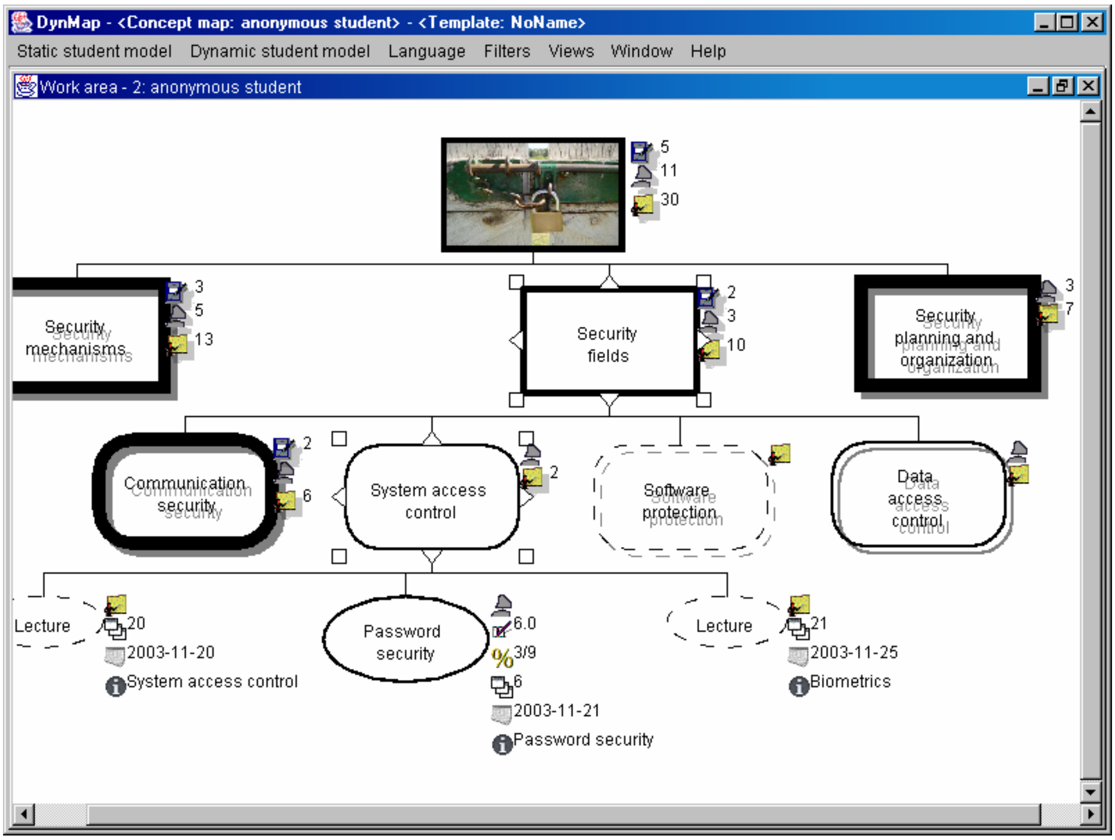


Figure 2.15: OSM in DynMap. Adapted from Rueda et al. [2006].

Figure 2.16 is a screen shot of concept map used in Flexi-OLM, which shows an example of presenting student's knowledge using predefined map. Topics are connected by one-way arrows, specifying their relationships. The same color scale is used to indicate user's understanding of a topic [Mabbott and Bull, 2006]. This concept map tends to be more complex than the topic hierarchy introduced in Figure 2.13. It delivers intricate domain structures and relationship among domain topics to viewers.

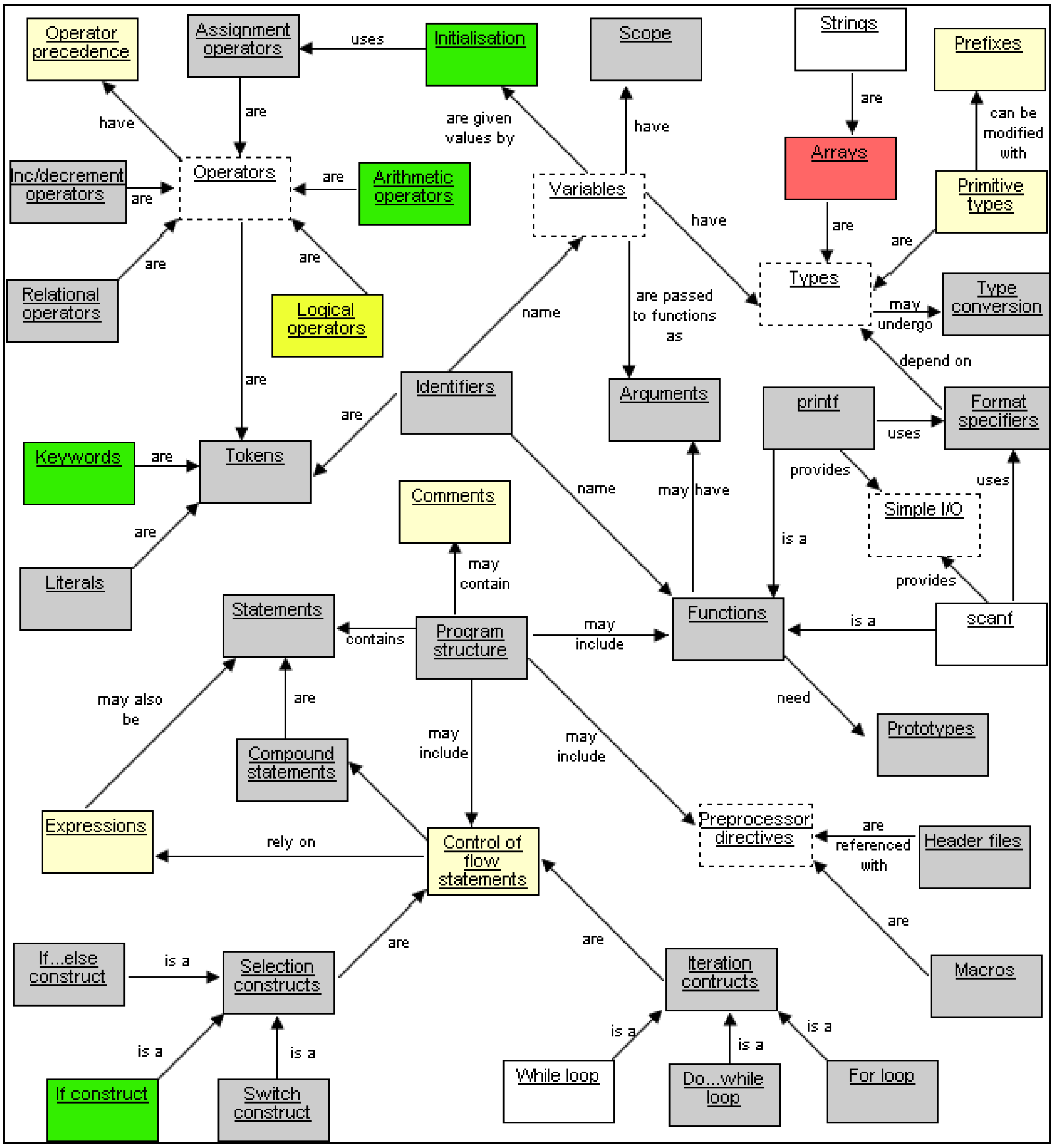
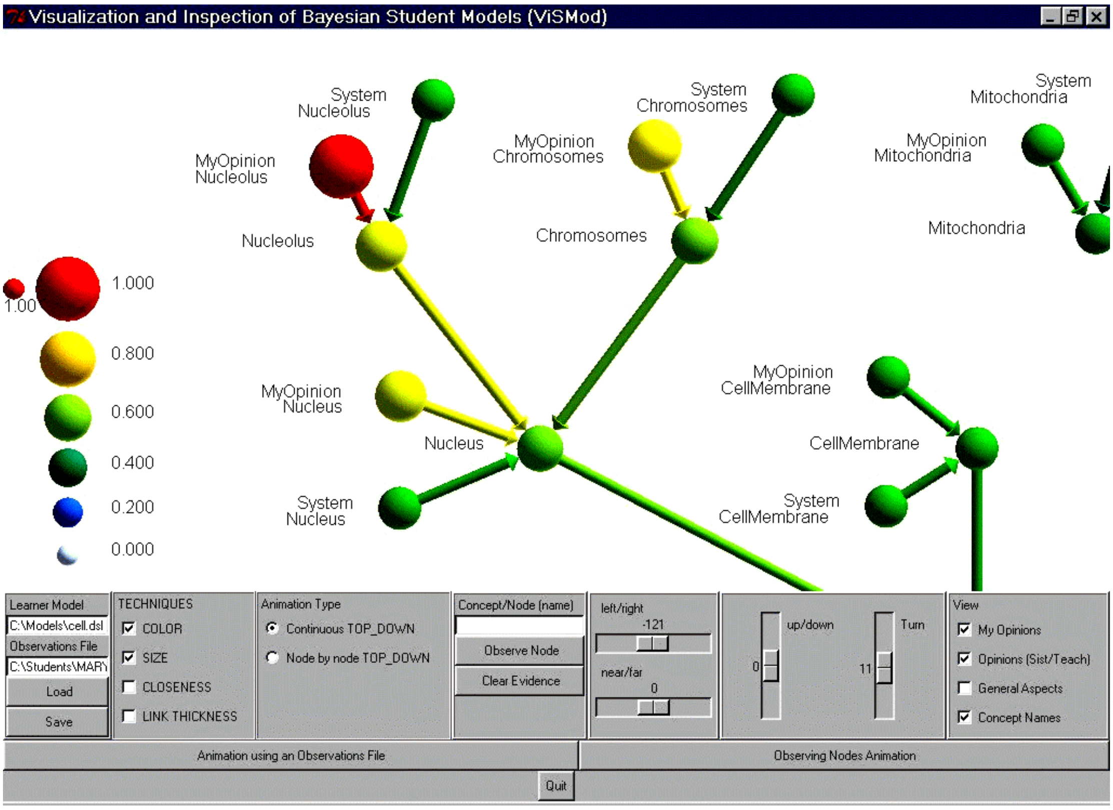


Figure 2.16: Concept map in Flexi-OLM. Adapted from Bull and Mabbott [2007].

Concept maps seem to be more complicated than other presentation styles. They may be very difficult to design and to understand. However, in a map view, students are able to see the differences in knowledge level at a glance. They may be distracted by the overall structure of concepts in the domain, but this may turn to be an advantage if one of the aims for OSM is encouraging students to learn such structure [Mabbott and Bull, 2007]. Students are also likely to spend time on exploring the areas with insufficient data. Consequently, the system may collect more data about students and then build better models.

2.3.7 Bayesian Networks

Zapata-Rivera and Greer [2001] have developed VisMod, an interactive visualization tool for Bayesian learner models. As displayed in Figure 2.17, learners and teachers can visualize the model using various techniques, such as color, size, closeness, link thickness, and animation.


Visualization and Inspection of Bayesian Student Models (ViSMod)

Learner Model: C:\Models\cell.dsl
Observations File: C:\Students\MARY

TECHNIQUES
☑ COLOR
☑ SIZE
☐ CLOSENESS
☐ LINK THICKNESS

Animation Type
● Continuous TOP_DOWN
○ Node by node TOP_DOWN

Concept/Node (name):

left/right -121
near/far 0
up/down 0
Turn 11

View
☑ My Opinions
☑ Opinions (Sist/Teach)
☐ General Aspects
☑ Concept Names

Figure 2.17: A fragment of Bayesian student model in VisMod. Adapted from Zapata-Rivera and Greer [2001].

VisMod is able to capture information about learner’s knowledge, learning preferences, or even social aspects of learning [Zapata-Rivera and Greer, 2001]. It aims to not only support students’ knowledge awareness but also refine learner models as students interact with their own models. However, in some domains, using Bayesian networks for OSM can be very complicated for both developers and learners.

2.3.8 Multiple Presentations for OSM

In the previous sections, we have introduced a wide variety of presentation styles for OSM. It can be seen that each of them has pros and cons, and therefore is suitable for different situations in terms of domains, system types, educational goals, and students' knowledge levels. If a system includes multiple views for its OSM, it may be more preferable for both educators and learners [Mabbott and Bull, 2004]. Previous research has raised the topic of multi-view OSM and has given some recommendations [Mabbott and Bull, 2007]. Based on their findings, several considerations should be given when offering students different views of student model.

First of all, it has been noticed that users do have different preferences for views to use. They may find some presentations more useful than others, and they may have different opinions about a particular presentation. For example, an alphabetically ranked list of concepts may be good when students want to search for certain concepts. For another example, since students have different levels of understanding about a domain, some of them may like concept map because it clearly shows where the problems are, whereas some others may be less inclined to use it due to its complexity.

The second consideration is that different purposes of OSM can be achieved using multiple presentations. Usually, the initial purpose of opening up student model is to support students' self-reflection and planning. Some presentations are appropriate for this purpose, such as a list of concepts ranked by level of knowledge. However, the usage of OSM is not limited to this. For example, another purpose may be allowing students to provide information to the areas that have insufficient data, so that more accurate models can be built. In this case, a concept map may work well due to its visualization of these areas.

Thirdly, the requirements of instructional domain should be taken into account. For instance, in some domains, designing a concept map or Bayesian network may be infeasible, whereas in some other domains, concept hierarchy may be unavailable.

Last but not least, students' focus of attention in presentation styles should also be considered. For example, if they find some formats particularly difficult, they are likely to switch to other views. If they are offered too many choices, they may scan each view very quickly and therefore neglect some important aspects.

CHAPTER 3

Research Hypotheses and Design

Chapter 3 starts with presenting our research objectives, including what we have aimed to discover from the research, and what hypotheses we have tried to prove. The chapter also demonstrates the processes of designing new views for the OSM in EER-Tutor. Firstly, we give an overview of the design for multi-view OSM; and then we introduce each of the new views in details.

3.1 Research Goals

This research has been aimed to achieve the following specific goals:

- Perform an intensive study of existing formats for visualizing student models in a variety of educational systems, in order to find out their advantages and disadvantages as well as their suitability in different situations.
- Analyze the current status of OSM in EER-Tutor, including how students are actually modeled, how student models are presented to users, and how effective the current presentation is.
- Select a set of most suitable views for OSM in EER-Tutor according to the characteristics of the domain and its students, and the requirements of the system.
- Design each of the selected views within the EER domain; so as to present domain concepts in appropriate ways in terms of layouts and meaningfulness.

- Implement these designs in the actual tutoring system, including the development of each view based on the underlying details of student models in EER-Tutor, the integration of multiple views into the system, and the implementation of interface controls that allow students to choose between different views.
- Evaluate the resulting EER-Tutor and its OSM by conducting an experiment that provides both subjective and objective results; analyze the results and then draw our conclusion.

3.2 Research Hypotheses

Before undertaking this research, we have made two main hypotheses that correspond to our research goals. Firstly, users have individual preferences of presentations for OSM in EER-Tutor. Secondly, the use of multi-view in OSM helps students with their learning in the tutoring system.

Previous research has pointed out that users do have preferences for presentation styles of student model. Mabbott and Bull [2007] have adapted their system Flexi-OLM, a multi-view open leaner model for the domain of C programming, to enable students to select their preferred presentations. The results have suggested that learners prefer a diversity of views for learner model and they benefit from the flexibility of using different layouts. In another study, Bull and her colleague have tracked learners' visual attention while using multiple views in Flexi-OLM. They have found that students visually attend to different views of leaner model [Bull et al., 2007]. Back to the research done in the ICTG, Hartley and Mitrovic [2002] have evaluated a two-level OSM used in one of our ITSs, KERMIT. The student model has been presented in two ways: one as a simple set of skill meters, the other as a hierarchy containing more details (see Subsection 2.3.5). Results have shown that students have appreciated having access to such OSM.

When students are offered several types of views, they can choose the ones they feel most comfortable to work with. Consequently, they can understand their student models better, as well as improve their meta-cognitive skills and even domain knowledge as much as they can. Such good understanding of student models may maximize the benefits of OSM. As we introduced before, the most important usage of OSM is to enhance students' abilities in self-assessment. Also, studies have shown that the development of students' meta-cognitive skills can result in better problem solving and better learning [Swanson, 1990]. Therefore, starting with an extended OSM in EER-Tutor, and then leading to the improvements of students' abilities in meta-cognition, we finally want to see the progress in their learning of the instructional domain.

This research is the first implementation of multiple views of student models in a constraint-based ITS. At the end the research, we aim to check whether students have individual differences in using these views, and whether the use of such OSM eventually lead to positive effects on students' learning processes.

3.3 Design

In order to implement multiple presentations for OSM in EER-Tutor, the first thing is to design a set of views that are appropriate for the domain and the system. Each view should represent EER domain at certain level. Together they should form a multi-layered OSM that can summarize the domain and analyze students precisely. These selected views need to be sketched out according to the knowledge base. For each view, it has to be decided what concepts should be included, what relationships should be presented, and how they should be visualized.

3.3.1 Overview

We have done an intensive study of OSM presentations used in existing educational systems, as summarized in Chapter 2. These presentation styles have been compared to see their strengths and possible weaknesses. Now we need to relate them to EER domain and our tutoring system, and decide which ones to include.

First of all, we need a simple OSM format that can be understood easily by every user. Originally, EER-Tutor already includes skill meters. Although they are just a high-level summary of the domain and represent only a small number of domain topics, they have been proved useful in other ITSs in terms of helping students with self-assessment [Mitrovic and Martin, 2007]. The successful use of skill meters in KERMIT, which is the predecessor of EER-Tutor, has encouraged us to keep this format. It is the simplest view among all the views. The brevity allows students to quickly comprehend meanings behind the view. The limitations have to be complemented by other views in the OSM.

Beyond skill meters, we need some views with higher complexities. That means, they should present more information about user and the domain than what skill meters do. We have finally chosen concept list, concept hierarchy and concept map, which are discussed respectively in the following sections. These views, along with skill meters, are not proposed to be shown all together to end users. We intend to present only one of them at a certain point of time. Students should be able to freely access the views they prefer and switch between views at any time they like.

3.3.2 Concept List

Since skill meters only display eight major domain topics (as shown in Figure 2.5) and there are a number of topics and sub-topics in EER domain, it is necessary to have a format of student model that shows all the concepts being taught in the tutoring system. In line with the layout of skill meters, we have decided to use a concept list rather than a text summary. The concept list is extended from skill meters. It represents a full list of domain concepts and remains a consistent looking with skill meters. Each concept is still associated with a coloured progress bar and two percentages: one for coverage of knowledge, the other one for correctly learned knowledge. For students’ convenience, these concepts are ranked by the order they are being introduced into the corresponding course. As the concept list adopts the appearance of skill meters, it should be easy to understand by both previous and future users of the system.

3.3.3 Concept Hierarchy

The views so far do not visualize the structure of the domain. They merely display partial or complete lists of domain concepts. In skill meters and concept list, all the concepts are treated equally, and their relationships are omitted. However, concepts in EER domain have close associations with each other. These associations can be illustrated using a concept hierarchy.

Recall that concept hierarchy is a tree view of OSM, in which all the domain concepts are categorized into branches and nodes. By including this kind of presentation style, we want to provide students with information about the structure of the EER domain. While using it, students may learn not only about their own performance but also some knowledge regarding to the generalization/specialization relationships among domain topics and concepts. Also, students may be able to locate a category of concepts that they are good or not good at. The understanding of this view may require some prior knowledge about the domain. Therefore, this concept hierarchy is considered to suit advanced students who want to gain an overall image of the knowledge base.

As shown in Figure 3.1, we have developed a hierarchy view for all the EER concepts, which can be used to represent OSM in the tutoring system. In this diagram, each box indicates a domain concept, and each arrow shows a top-down relationship between two concepts. All the concepts are categorized according to their meaning. Together they build different levels of the tree. Same as in other views, each concept in the hierarchy has been planned to have two percentages that tell about user’s knowledge levels.

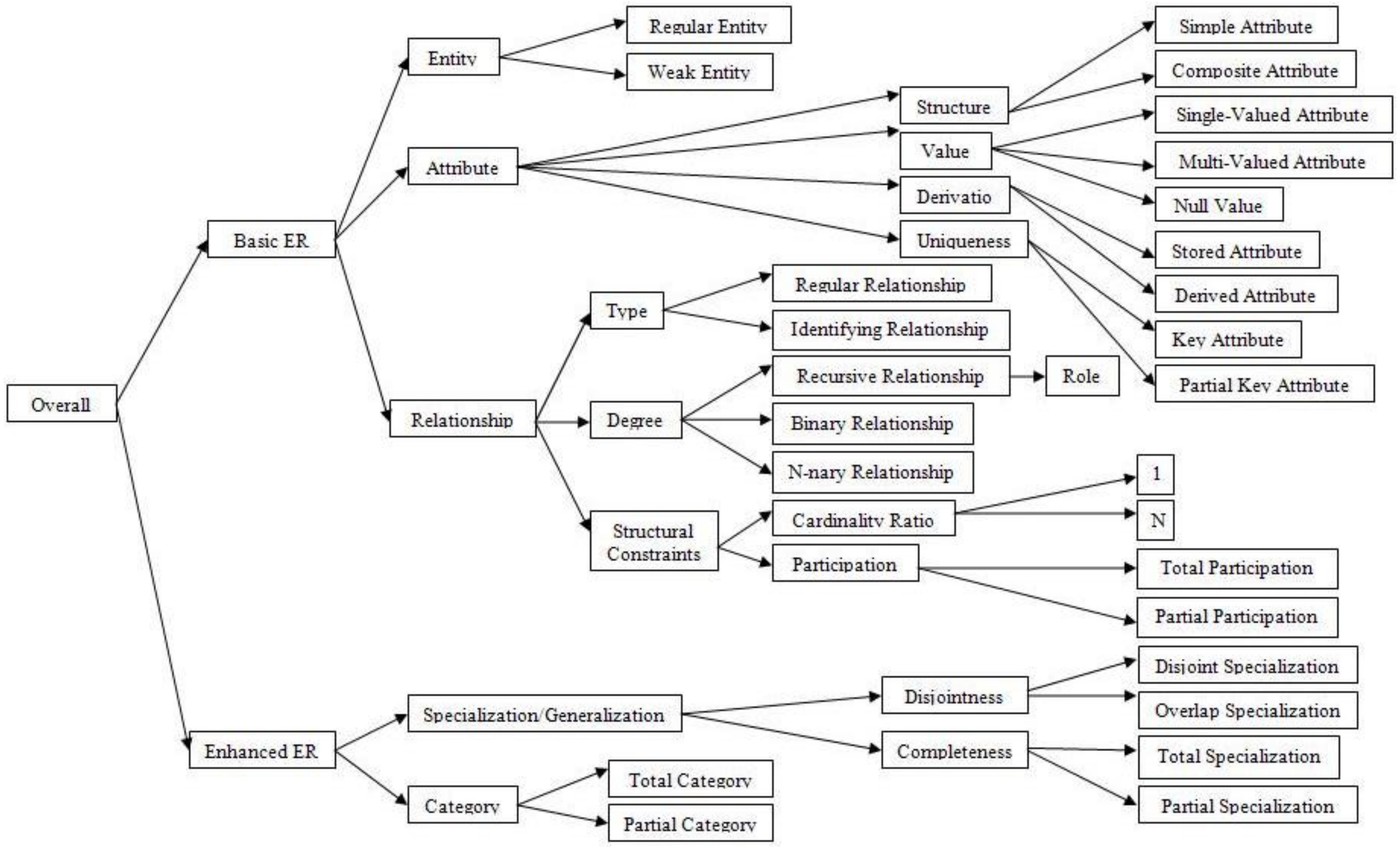


Figure 3.1: Concept hierarchy of EER domain.

3.3.4 Concept Map

Although the concept hierarchy presents the structure of the whole EER domain and categorizes all the topics and concepts, there are still relationships that cannot be represented in a tree view. Actually, the concepts and relationships among EER concepts can be really intricate. We need a map-like view to visualize them. Moreover, since our purposes of using OSM are that students do not only reflect on their own learning processes but also gain more knowledge about the domain, a well designed concept map is a necessary member of the multi-view OSM. Concept map is usually hard to design and understand since it is one of the most complex presentation techniques. However, the extra information being delivered by concept map can be a very interesting challenge for advanced students.

As a popular presentation method of OSM, concept maps have been implemented in many educational systems with various layouts. In our case, we have designed an EER diagram to represent the concept map of EER domain, as shown in Figure 3.2. Table 3.1 provides a summary of ER notation as well [Elmasri and Navathe, 2003]. EER diagrams are generally used for modelling databases, but it also works well in illustrating complicated relationships of EER concepts. All the domain concepts are drawn in the diagram, either as entities or as attributes. Their relationships being displayed in this view are much more comprehensive than the top-down relationships being shown in the concept hierarchy. Each concept has also been planned to have two percentages corresponding to taught and learned knowledge.

By examining this EER diagram of EER concepts, students have a chance to enhance their meta-cognitive skills, learn domain concepts, as well as study an example of EER modelling, as EER modelling is the primary teaching goal of EER-Tutor. Moreover, when a concept has not been well learned, other concepts close to it on the map are also likely to have knowledge gaps, so students may be able to find out the areas they are having problems with.

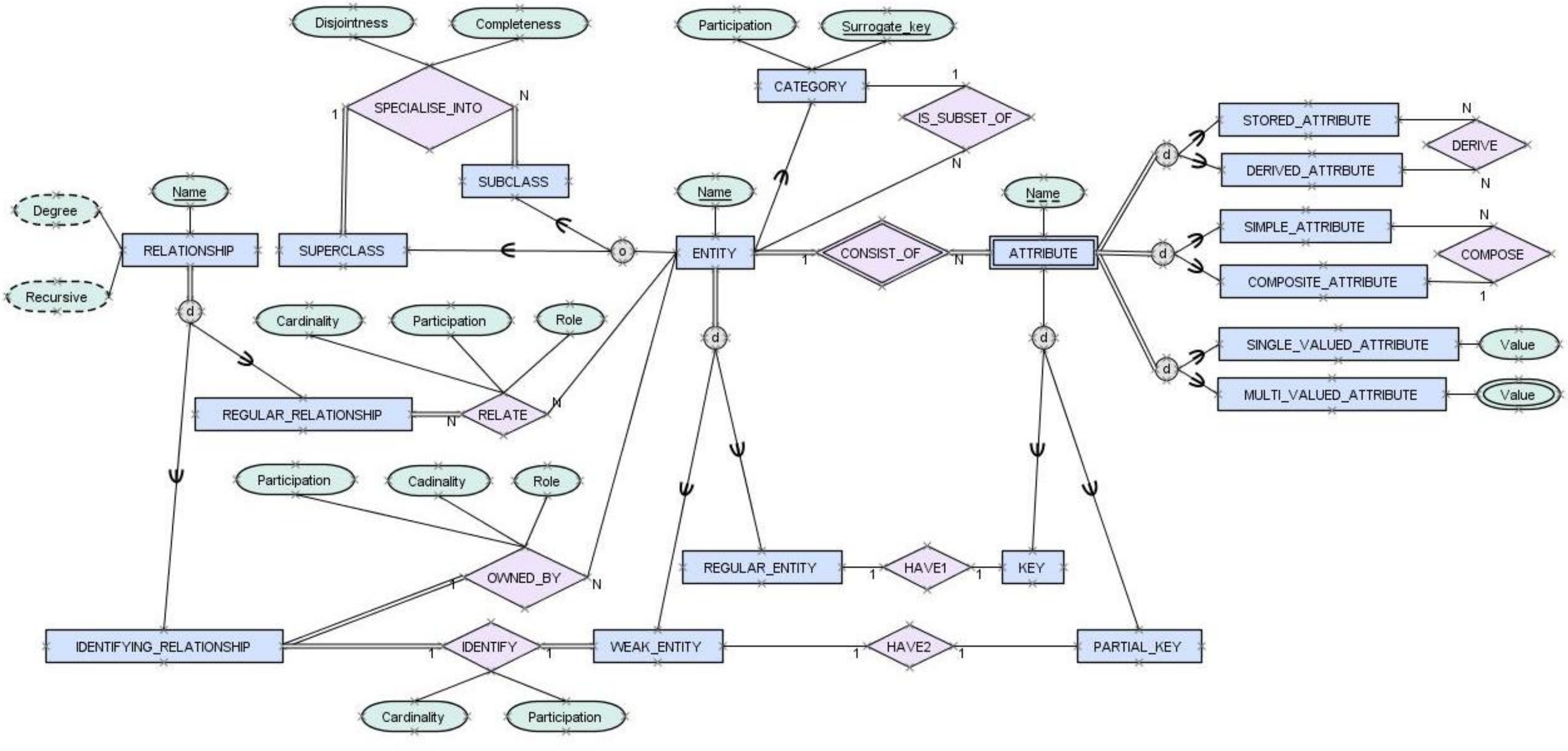
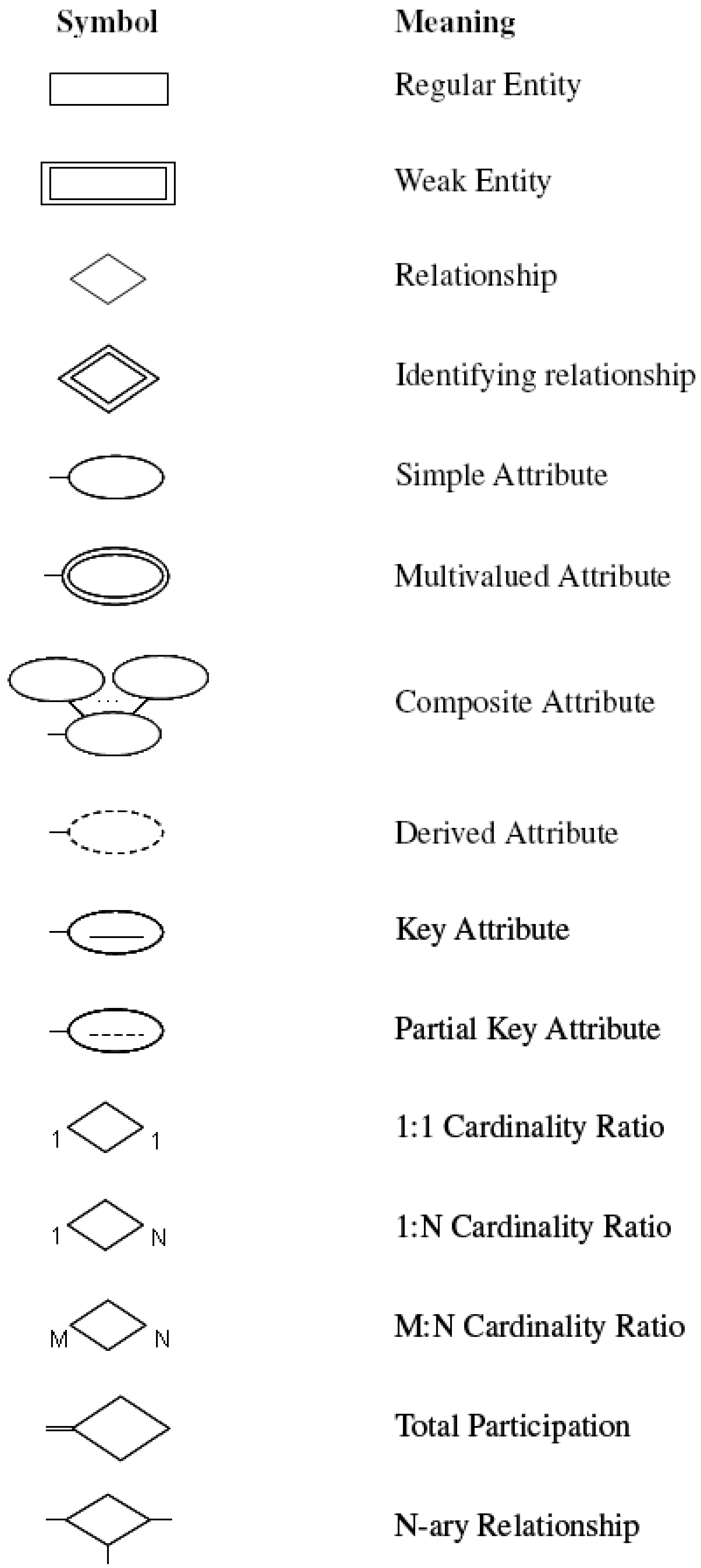


Figure 3.2: EER domain concepts presented as an EER diagram



Symbol	Meaning
	Regular Entity
	Weak Entity
	Relationship
	Identifying relationship
	Simple Attribute
	Multivalued Attribute
...	Composite Attribute
	Derived Attribute
	Key Attribute
	Partial Key Attribute
1 1	1:1 Cardinality Ratio
1 N	1:N Cardinality Ratio
M N	M:N Cardinality Ratio
	Total Participation
	N-ary Relationship

Table 3.1: Summary of ER notions. Adapted from [Elmasri and Navathe, 2003].

CHAPTER 4

Implementation

Once the new views for OSM in EER-tutor have been designed, the next phase is the actual implementation of such OSM in the system. To achieve that, the original EER-Tutor has been well studied, with some parts of the system being particularly examined, such as the semantic and syntax constraints. As the core of the research, issues related to the OSM itself has also been learned, including how student models are set up, how they are updated, and how they are made viewable to students. After that, we have implemented the new views in line with the original system.

In this chapter, we firstly introduce how the constraints in the system are associated with the concepts being displayed in OSM; and then we explain how the percentages for covered and learned knowledge are calculated; finally the interface of the extended OSM are addressed in the last section.

4.1 Mapping between Constraints and Concepts

In Section 2.2 we have looked at the current implementation of OSM in EER-Tutor, which is in the format of skill meters. There are only eight major concepts shown to students as progress bars. However, in the multi-view version of OSM, there are more than thirty categories and concepts available for viewing. To fit our new design into the current system, each concept being displayed in the new presentations must rely on a list of related constraints, and vice versa, each constraint can affect a set of concepts. This mapping implies that learning or failing to learn a constraint updates the two percentages of associated concepts, and then the violation and satisfaction of all the constraints relevant to a particular domain concept can tell how much the student has mastered that concept.

To achieve this mapping, we need to find out the existing correlations between constraints and domain concepts. Originally, each constraint has been bonded with a concept by labelling the constraint with a concept name. A constraint always belongs to only one concept. The following example shows constraint 14, which checks whether a student has correctly identified weak entities. In the second last line, the constraint has been labelled as “entity types”:

```
(14
    "Check your regular entity types. Some of them should be
represented as weak entity types."
    (and (match IS ENTITIES (?* "@" ?tag ?l1 "weak" ?*))
        (match SS ENTITIES (?* "@" ?tag ?l2 ?type ?*)))
    (test SS ("weak" ?type))
    "entity types"
    (?tag))
```

In the new version of OSM, such a constraint does not only belong to “Entity” category, but is considered to affect “Regular Entity” and “Weak Entity” as well since the student who violates this constraint has confusion about these two concepts. This change causes the constraint being associated with more than one concept. For the whole concept set and constraint set, their 1:N relationships become M:N relationships. In this case, we cannot simply change the label of constraint 14 from “entity type” to another concept name.

To solve this problem, we have decided to use another way to represent the relationships between constraints and concepts. All of their correlations have been hard-coded in a hash-table-like structure, which can be used later for setting up and updating student models. The table contains forty rows, each of which represents a domain topic or concept. A row consists of two parts, with the first part being the name of that concept and the second being a list of constraint numbers. When a solution of a particular problem is submitted, it is checked against all the relevant constraints for that problem. Each of these constraints belongs to several concepts according to the hash-table. Consequently, the covered and learned percentages of these concepts are going to be changed depending on the correctness of the solution.

4.2 Summary of Learning

One important thing that has not been addressed yet is how the percentages are actually calculated. In order to answer this question, we firstly need to look at what is actually inside a student model. The tutoring system saves each student's constraint usage into a model file. The file shows a list of constraints that have been used by the corresponding student as well as their violation and satisfaction history. The example below is part of a model file:

```
(61 0 (1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 1 1 1 1) 4/5)
(60 0 (1  1 1 1 0 0 0 0 0 0 0 1 1 1 0 0 0 11 1 1) 4/5)
(51 0 (1 1 1) 1)
(50 0 (1 1 11 1 1 1 1 1 1 1 1 1 1 1 1) 1)
(209 0 (1 1 1 1 1 1) 1)
(16 0 (1 0 0 0 1 0) 1/5)
(14_1 0 (1 1 1 1 1 1) 1) (14 0 (1 1 0) 2/3)
(45 0 (1 1 1 1 1 1 1 0 1 1 1 0 1 1 0 1 1 0 0 0 1 1 1 0) 3/5)
(33 0 (1 1 0) 2/3)
(21 0 (1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1) 1)
...
```

The first number in a row indicates the constraint number. Those zeros and ones enclosed inside a pair of parentheses record the history of usage for that constraint, with one meaning satisfaction and zero meaning violation. The last number of the row is the correctness ratio for that constraint, which is decided by the number of correct attempts in last five attempts performed by the student.

The formulas used for skill meters have been continuously applied to all the new categories and concepts in the extended OSM:

$$\text{Percentage of coverage} = \frac{\text{Number of constraints relevant to the concept in student model}}{\text{Number of constraints relevant to the concept in the knowledge base}}$$

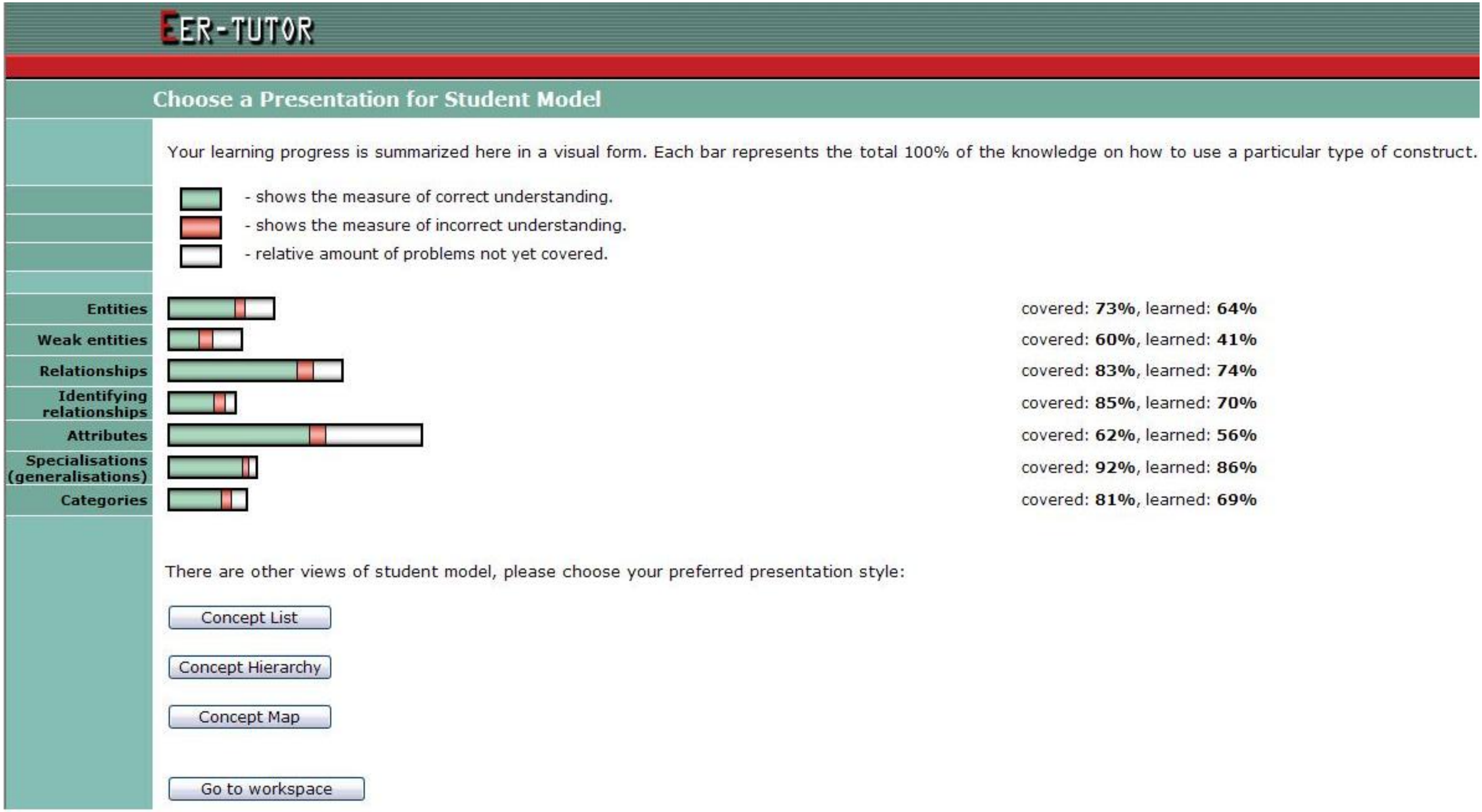
$$\text{Percentage of correctness} = \frac{\text{Sum of correctness ratios for constrains relevant to the concept in student model}}{\text{Number of constraints relevant to the concept in the student model}}$$

Now the system is able to approximately calculate the relative amount of knowledge for each concept, how much knowledge has already been taught for each concept, and how a student performs on each concept. This data is ready to be displayed to students in any formats.

4.3 Interface

In the new version of OSM in EER-Tutor, we include a variety of presentations, but student is allowed to view only one of them at any point of time. Therefore, when student requests his/her student model, we only show skill meters since it is the easiest understandable view, but there are several new buttons at the bottom on the same page, which allows student choosing among new views or going back to the main working space, as shown in Figure 4.1. Moreover, in each presentation page, there are two buttons under the view (see Figure 4.2), one for going back to the previous page for skill meters, the other for returning to the main page of the system.

The concept list demonstrated in Figure 4.2 has been directly extended from skill meters. It adopts the layout of skill meters so that previous students may find it familiar. All the domain concepts that we want to teach have been included in this list, with different lengths indicating their relative amounts of knowledge and different percentages according to the calculation done at previous stages.



Choose a Presentation for Student Model

Your learning progress is summarized here in a visual form. Each bar represents the total 100% of the knowledge on how to use a particular type of construct.

🟩 - shows the measure of correct understanding.

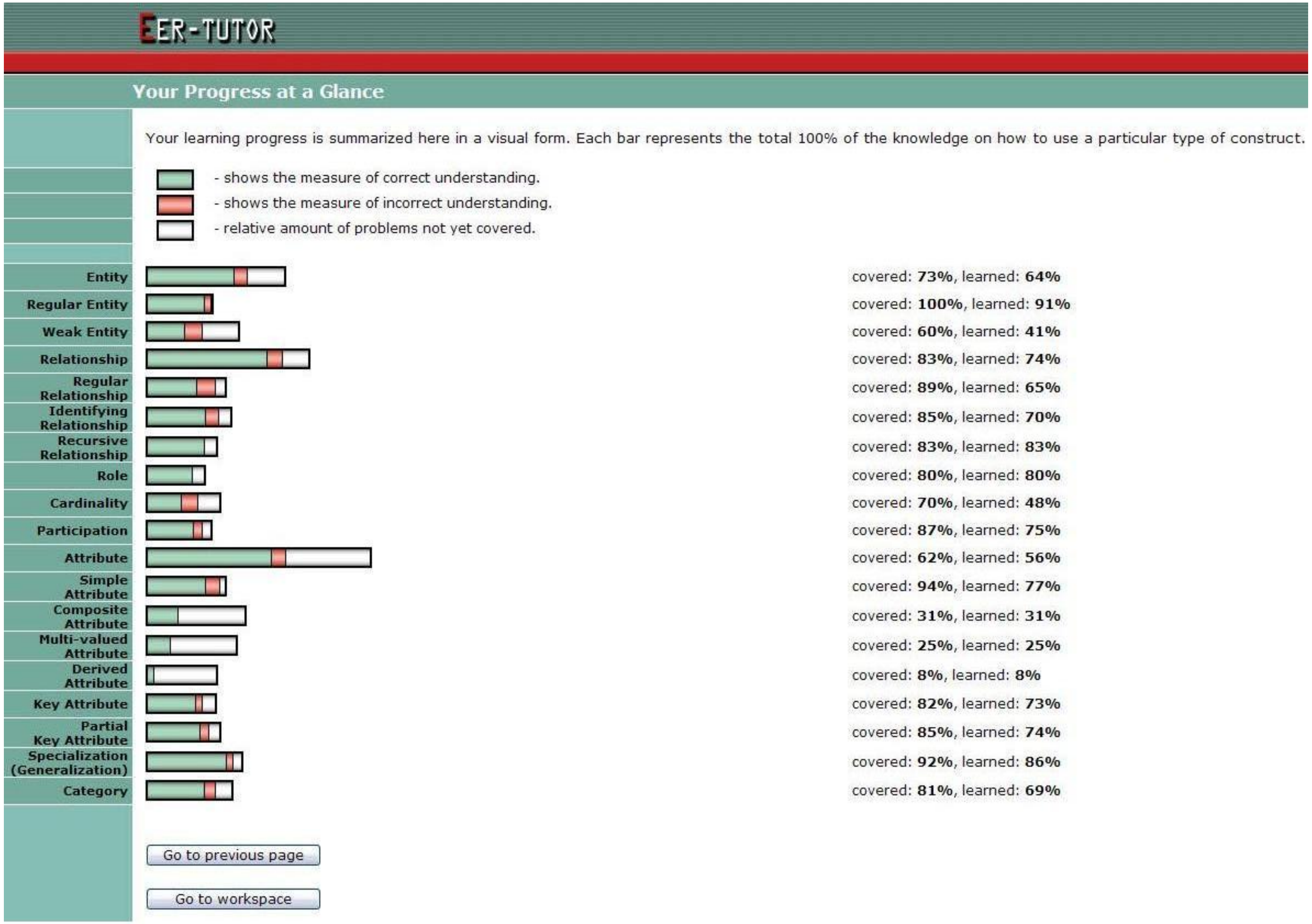
🟥 - shows the measure of incorrect understanding.

□ - relative amount of problems not yet covered.

Entities	covered: **73%**, learned: **64%**
Weak entities	covered: **60%**, learned: **41%**
Relationships	covered: **83%**, learned: **74%**
Identifying relationships	covered: **85%**, learned: **70%**
Attributes	covered: **62%**, learned: **56%**
Specialisations (generalisations)	covered: **92%**, learned: **86%**
Categories	covered: **81%**, learned: **69%**

There are other views of student model, please choose your preferred presentation style:

Figure 4.1: The main page of OSM in EER-Tutor.



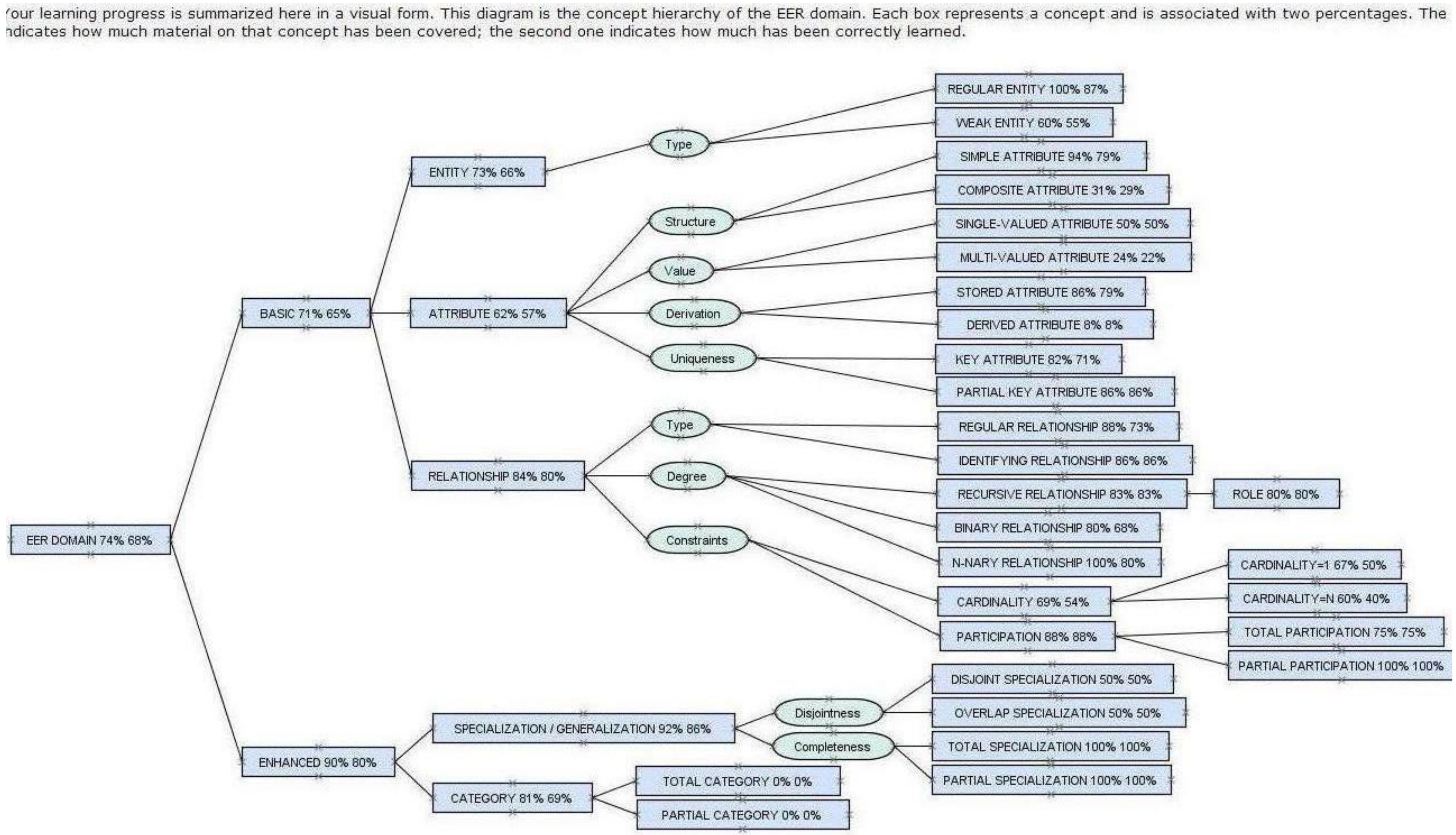
Your Progress at a Glance

Your learning progress is summarized here in a visual form. Each bar represents the total 100% of the knowledge on how to use a particular type of construct.

🟩 - shows the measure of correct understanding.
🟥 - shows the measure of incorrect understanding.
□ - relative amount of problems not yet covered.

Entity	covered: **73%**, learned: **64%**
Regular Entity	covered: **100%**, learned: **91%**
Weak Entity	covered: **60%**, learned: **41%**
Relationship	covered: **83%**, learned: **74%**
Regular Relationship	covered: **89%**, learned: **65%**
Identifying Relationship	covered: **85%**, learned: **70%**
Recursive Relationship	covered: **83%**, learned: **83%**
Role	covered: **80%**, learned: **80%**
Cardinality	covered: **70%**, learned: **48%**
Participation	covered: **87%**, learned: **75%**
Attribute	covered: **62%**, learned: **56%**
Simple Attribute	covered: **94%**, learned: **77%**
Composite Attribute	covered: **31%**, learned: **31%**
Multi-valued Attribute	covered: **25%**, learned: **25%**
Derived Attribute	covered: **8%**, learned: **8%**
Key Attribute	covered: **82%**, learned: **73%**
Partial Key Attribute	covered: **85%**, learned: **74%**
Specialization (Generalization)	covered: **92%**, learned: **86%**
Category	covered: **81%**, learned: **69%**

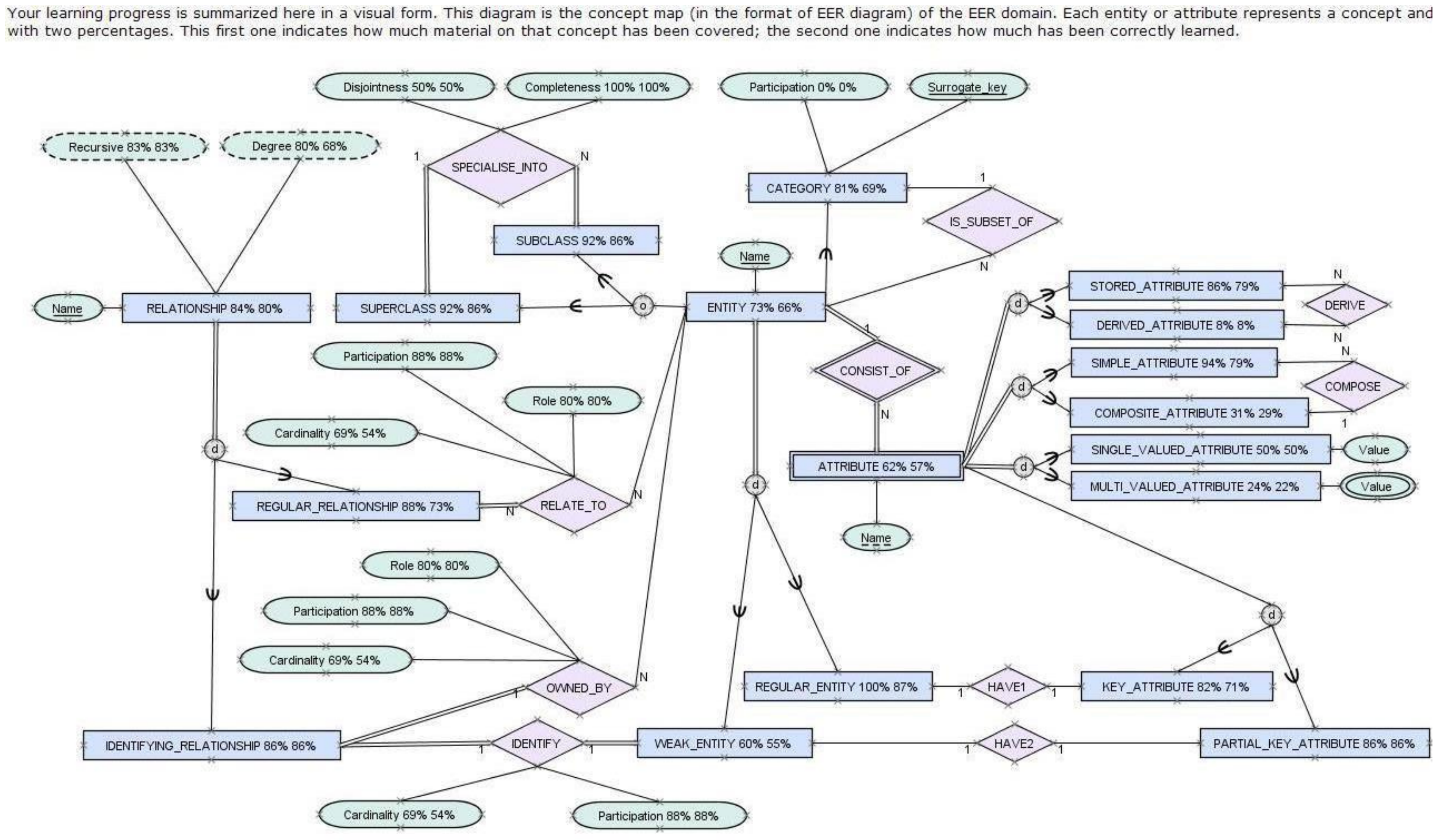
Figure 4.2: Concept list of the OSM in EER-Tutor



Your learning progress is summarized here in a visual form. This diagram is the concept hierarchy of the EER domain. Each box represents a concept and is associated with two percentages. The
indicates how much material on that concept has been covered; the second one indicates how much has been correctly learned.

EER DOMAIN 74% 68%
BASIC 71% 65%
ENTITY 73% 66%
Type
REGULAR ENTITY 100% 87%
WEAK ENTITY 60% 55%
ATTRIBUTE 62% 57%
Structure
SIMPLE ATTRIBUTE 94% 79%
COMPOSITE ATTRIBUTE 31% 29%
Value
SINGLE-VALUED ATTRIBUTE 50% 50%
MULTI-VALUED ATTRIBUTE 24% 22%
Derivation
STORED ATTRIBUTE 86% 79%
DERIVED ATTRIBUTE 8% 8%
Uniqueness
KEY ATTRIBUTE 82% 71%
PARTIAL KEY ATTRIBUTE 86% 86%
RELATIONSHIP 84% 80%
Type
REGULAR RELATIONSHIP 88% 73%
IDENTIFYING RELATIONSHIP 86% 86%
Degree
RECURSIVE RELATIONSHIP 83% 83%
ROLE 80% 80%
BINARY RELATIONSHIP 80% 68%
N-NARY RELATIONSHIP 100% 80%
Constraints
CARDINALITY 69% 54%
CARDINALITY=1 67% 50%
CARDINALITY=N 60% 40%
PARTICIPATION 88% 88%
TOTAL PARTICIPATION 75% 75%
PARTIAL PARTICIPATION 100% 100%
ENHANCED 90% 80%
SPECIALIZATION / GENERALIZATION 92% 86%
Disjointness
DISJOINT SPECIALIZATION 50% 50%
OVERLAP SPECIALIZATION 50% 50%
Completeness
TOTAL SPECIALIZATION 100% 100%
PARTIAL SPECIALIZATION 100% 100%
CATEGORY 81% 69%
TOTAL CATEGORY 0% 0%
PARTIAL CATEGORY 0% 0%

Figure 4.3: Concept Hierarchy of the OSM in EER-Tutor.



Your learning progress is summarized here in a visual form. This diagram is the concept map (in the format of EER diagram) of the EER domain. Each entity or attribute represents a concept and with two percentages. This first one indicates how much material on that concept has been covered; the second one indicates how much has been correctly learned.

Figure 4.4: Concept map of the OSM in EER-Tutor.

The implementation of web interface for concept hierarchy and concept map is different. Since these two structures are much more complicated than the concept list, we have adopted a different approach to achieve it. Each of these diagrams has been drawn as an XML file with structures only. When a view is requested by student, its XML file is loaded, and then each of the concept objects in the XML file is filled with corresponding percentages. Finally a print method transforms the resulting XML file into a viewable web page for the student. The resulting diagrams for concept hierarchy and concept map are shown in Figure 4.3 and Figure 4.4 respectively. Each of the boxes in the concept hierarchy contains the percentages for covered and learned knowledge after the concept name, so does each of the major objects in the concept map.

CHAPTER 5

Evaluation

In March 2009, we conducted a study in order to evaluate our research hypotheses. The evaluation process consisted of both surveys and tests, since we wanted to collect not only subjective opinions but also objective data. The results of the evaluation have been collected and analyzed.

In this chapter, we first describe the processes of the evaluation study, followed by an introduction of necessary modifications done in the tutoring system. In Section 3 and Section 4, we give the results from this evaluation, followed by discussions.

5.1 Evaluation Description

Recall that there are mainly two things we want to find out from this research: firstly, we want to see how multiple presentations of OSM affects' student learning; secondly, we want to know what students' preferences are in terms of different presentation styles for OSM. In order to answer these questions, we have conducted an evaluation study involving pre/post tests and a questionnaire. From the pre-test and post-test, we intend to see whether students' learning has been positively affected by the extended system, whereas from the questionnaire we hope to hear students' comments on different aspects of the multi-view OSM.

The study was run with students enrolled in COSC226 Introduction to Databases of the University of Canterbury in Semester one, 2009. The participant used EER-Tutor in their regular lab sessions during the third week of the course, by which they had already studied major concepts of the ER domain and begun to learn about the Enhanced ER model. In the last lecture before the study, they were given a brief introduction to EER-Tutor and its OSM. At the beginning of the first lab session, the purpose and process of the evaluation study were also addressed.

The first time the participants logged into EER-Tutor, they were randomly divided into two groups by the system: the control group and the experimental group. The control group was given access to the original EER-Tutor, which presented student model using only skill meters. In contrast, the experimental group used the EER-Tutor with extended OSM, which provided a set of views as described in Chapter 4. Students were not aware of the existence of the two groups.

The participants were presented with an information page, which introduced the system. The information page also explained the student model, discussing what it would be used for and how to access it. Since students belonged to different groups, they saw different introductions about the OSM. After the information page, students proceeded to perform a pre-test online before they could do anything else. The test contained a set of simple multi-choice questions about data modelling using EER notation. Once students completed the pre-test, they were brought to the main working space, where they could start working on the problems presented by the system.

The participants did not only use the tutoring system during their lab sessions. They were allowed to access EER-Tutor for three weeks. At the end of this period, the participants were asked to do a post-test on paper. The post-test had same number of questions about EER domain and same level of difficulty as the pre-test. After the post test, students were also given questionnaires to fill out. The questions related to different aspects of OSM in EER-Tutor, such as the usefulness and difficulty of the OSM. The participants could rate their satisfaction with various aspects of the OSM, and there were open-ended questions for them to leave any comments. In addition, the experimental group was given extra questions about students' preferences over different presentation styles of the multi-view OSM.

5.2 System Modifications

In order to run the evaluation study, several things in the tutoring system have been modified to cater our needs. We have created two modes in the system, representing the original and the extended version of EER-Tutor. The system has been extended to alternatively assign new user into each group by marking the user with the corresponding mode. Also, from the time that a user logs into EER-Tutor, all the actions of the user are recorded, especially the interactions with the OSM. For the experimental group, things like what time user enters a particular view or switches from one view to another have been kept in student logs.

To promote the use of OSM, we have also adapted the system to show student model immediately after user's log in. That means, when a returning student logs in, the student is brought to the OSM page directly. The control group sees skill meters and a button that leads them to the main working space; whereas the experimental group also sees skill meters and the same button, but with extra buttons linking to other views of the OSM (as shown in Figure 4.1). By doing this, we draw student's attention to the summary of their learning up to the last time they have used the system before they start solving any problems.

5.3 Test Results and Discussion

In 2009 there were 102 students enrolled in COSC226, 94 of who participated in the study. The pre-test was administered on-line when students logged into EER-Tutor for the first time, and some students have not attempted any questions included in it. We have excluded the data collected about those students from the analyses, and therefore 84 students appears in Table 5.1, which reports some statistics from the study.

	Control Group	Experimental Group	p
Participants	42	44	
Pre-test mean (%)	60 (24)	53 (25)	ns
# Post-test completed	19	17	
Pre-test for the subset (%)	63 (24)	57 (18)	ns
Post-test mean (%)	69 (15)	68 (16)	ns
Interaction time	5:53:29	5:26:57	ns
OSM viewed - total	18.6 (13.1)	15.3 (10.9)	ns
OSM viewed - self	2.2 (2.4)	2.5 (3.7)	ns

Table 5.1: Summary of students' interaction with EER-Tutor (standard deviations reported in parentheses).

There are seven questions in the pre-test (given in Appendix A), each of which contributes one mark to the total score of 7. There was no significant difference on the pre-test performance of the two groups. 36 students sat both pre-test and post-test (19 students in the control group and 17 in the experimental group). They all have done the questionnaires as well. Therefore, our initial analyses were based on those 36 students. We compared the performance on the pre-test for those students, and again there was no significant difference (reported in Table 5.1 in the "Pre-test for the subset" row).

The post test also has seven questions (given in Appendix A), and both groups obtained very similar results. A paired-t-test showed no significant improvement between pre- and post-test for the control group participants. However, the improvement was significant for the experimental group (p=0.04).

Control group participants have viewed the OSM (i.e. skill meters only) on average 18.6 times throughout their interaction, while the experimental group students have viewed various types of OSM 15.3 times. Table 5.1 also reports the number of times the participants explicitly requested to view the OSM (the "OSM viewed – self" row). The experimental group have requested to view the OSM 2.5 times on average, comparing to 2.2 times for the control group. There was no significant difference between the two groups on either the total times of viewing or the number of self-requested viewing of the OSM.

In Table 5.2, we show the average time (in seconds) and the number of times that the experimental group participants have viewed each presentation. The participants spent slightly more time (10.6 seconds on average) on concept hierarchy, compared to 8 seconds for concept list and 5.4 seconds for concept map. It also can be noticed that the standard deviations for time spent on concept list and concept hierarchy are high. The second column of the table shows the average number of times of opening each OSM presentation, which are similar for the three presentation styles.

	Viewing Time (Seconds)	No. of Viewings
Concept list	8.0 (23.6)	0.7 (1.1)
Concept hierarchy	10.6 (30.7)	0.6 (0.9)
Concept map	5.4 (13.3)	0.6 (0.7)

Table 5.2: Summary of the experimental group’s interaction with the OSM (standard deviations reported in parentheses).

We have also conducted a number of correlation tests for the data collected from the study. Table 5.3 presents the significant correlations between indicators for viewing the OSM and other test data. In the control group, as shown in Table 6.3 (a), the time students spent on the OSM (i.e. skill meters only) and the number of times they accessed the OSM, either the total or how many times they have requested to see the OSM, are all positively correlated with the total interaction time with the system. The number of problems solved by these participants is also significantly related to the viewing time and number of accesses of the OSM. Table 5.3 (b) displays the results of same tests for the experimental group. The time for viewing all the presentations in the OSM is highly correlated with the interaction time in the tutoring system. Again, there are also significant positive correlations between the number of solved problems and the use of the OSM, regardless which the type of views being used.

	Viewing Time	No. of Viewings (Total)	No. of Viewings (Self)
Interaction Time	0.379 (0.013)	0.761 (0.000)	0.383 (0.012)
# Problems Solved	0.389 (0.011)	0.462 (0.002)	0.462 (0.002)

(a) Control group

	Viewing Time	No. of Viewings (Total)	No. of Viewings (Self)
Interaction Time	0.383(0.009)	0.736(0.000)	0.150(0.326)
# Problems Solved	0.340(0.022)	0.443(0.002)	0.471(0.001)

(b) Experimental Group

Table 5.3: Correlations for the OSM (p-value reported in parentheses).

From these results we can see that students from both groups did regularly spend time on the OSM during their learning sessions. The more they stayed in the system, the more they explicitly requested to see the OSM. Moreover, it is also likely that they solved more problems when they used the OSM more often.

For the experimental group, we have also looked at how the viewing times of each new view are related to the number of times that participants used the OSM. Table 5.4 shows that the use of the concept hierarchy and the concept map is correlated with the total number of viewing the OSM. In the case that students opened the OSM themselves, all three OSM presentations are highly correlated with the viewing times. It means that students were likely to check the new views when they were self-motivated to use the OSM. In the last row of the table, we can also see that the viewing times are significantly correlated to the number of solved problems. Since the concept list and the concept hierarchy have got higher values (0.454 and 0.539

respectively) comparing to the concept map (0.299), they played more important roles in helping students with problem solving.

	No. of Viewings Concept List	No. of Viewings Concept Hierarchy	No. of Viewings Concept Map
Total	0.279(0.063)	0.413(0.005)	0.521(0.000)
Requested	0.579(0.000)	0.772(0.000)	0.455(0.002)
# Problems Solved	0.454(0.002)	0.539(0.000)	0.299(0.046)

Table 5.4: Correlations for different views in the OSM for experimental group (p-value reported in parentheses).

Finally, it is also interesting to find out there are correlations among three new presentations styles. As displayed in Table 5.5, the access times of these views are highly correlated to each other, with the highest to be 0.732 for concept list and concept hierarchy. It shows that students who used one of the views were likely to access the other views.

	Concept Hierarchy	Concept Map
Concept List	0.732(0.000)	0.550(0.000)
Concept Hierarchy		0.544(0.000)

Table 5.5: Correlations among the new views (p-value reported in parentheses).

All these data and statistical tests have shown us a picture of how the participants used the OSM to assist in their learning processes. Both groups took benefits from using the OSM, with the experimental group achieving a significant improvement. Students in the experimental group showed interests in using the new presentations of the OSM, and the use of them directly associated with students' problem solving results.

5.4 Questionnaire Results and Discussion

The questionnaire (given in Appendix B) consists of multi-choice questions and open questions. The first question asked whether they only viewed the OSM after each time they logged in as the system would promote it or they checked it at other time. Six out of nineteen (31.58%) students in the control group and eight out of seventeen (about 47.06%) students in the experimental group did check the OSM themselves. When they were asked the reasons for viewing the OSM, four students from the experimental group talked about their interests in finding out what information the OSM had. However, two students said the concept map was too hard to understand.

	Control Group	Experimental Group
Difficulty of the OSM	1.89 (0.76)	1.71 (0.61)
Usefulness of the OSM	3.18 (0.64)	3.21 (1.31)
Enjoyment of use	3.22 (0.94)	3.14 (1.10)

Table 5.6: Averages ratings for different aspects of the OSM (standard deviations reported in parentheses).

Students were also asked to rate the difficulty, usefulness and enjoyment of use for the OSM (as shown in Table 5.6) on the scale of 1 (“not at all”) to 5 (“very much”). The distribution of the rating for the difficulty is shown in Figure 5.1. Nearly half of the students, no matter which group they belong to, rated 2. There is no one giving a score over 3. Surprisingly, the experimental group has got a lower average rate of 1.71, comparing to 1.89 for the control group. Students were also asked why they gave such ratings to the OSM. They answered that the layout was clear and the explanation on each page was helpful. Several participants said that the OSM was initially difficult but once used becoming fine to understand. In addition, the experimental group was asked which presentation styles were relatively difficult to comprehend. More than half of the group thought the concept map being more difficult. Among them, three people also voted for concept hierarchy. One person wrote that the last two views were too complicated, especially the concept map, while the other student wrote "it was good that the concept map was itself presented as an EER diagram".

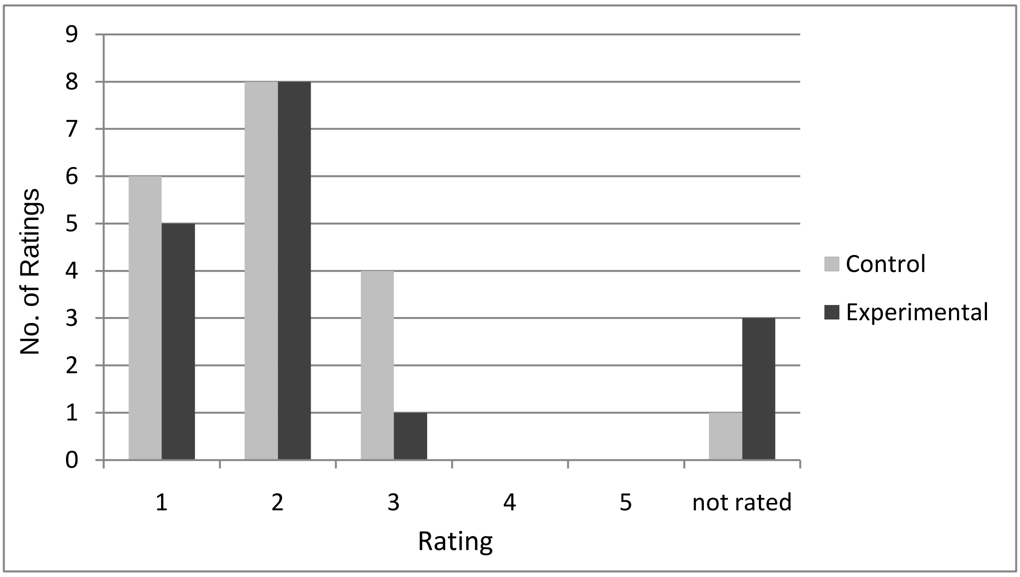


Figure 5.1: Ratings for difficulty of the OSM.

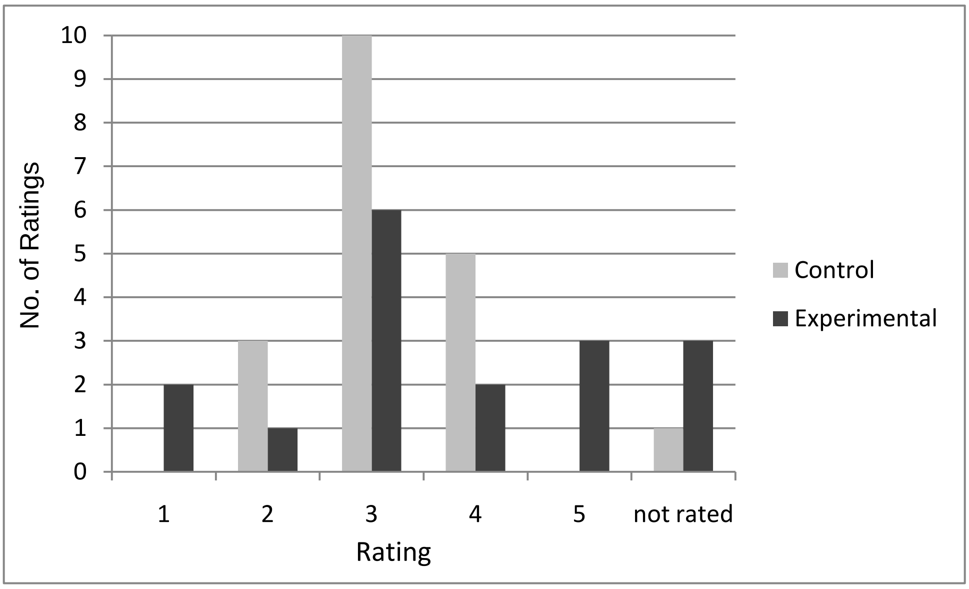


Figure 5.2: Ratings for usefulness of the OSM.

The following question is checking whether students considered the OSM being useful. This time (about) half of the students gave the neutral score 3, as illustrated in Figure 5.2. There is no one in the control group giving 1 or 5. However, it is interesting to see that opinions from the experimental group are polarized. Two people from the experimental group thought the OSM was not useful at all (scored 1), whereas three others claimed that it was very useful (scored 5). Overall, the control group rated 3.18 on average, which is slightly lower than 3.21 from the experimental group. When the experimental group was asked which views were useful, they voted for all the views except concept map. The participants also provided positive and negative feedback about the usefulness of the OSM. Seven students said that it showed what the student did and did not understand, and the details were appropriate. Three others doubted about the accuracy of student model and questioned about how everything was calculated. One student stated that the concept hierarchy was the best, as the concept map was hard to follow.

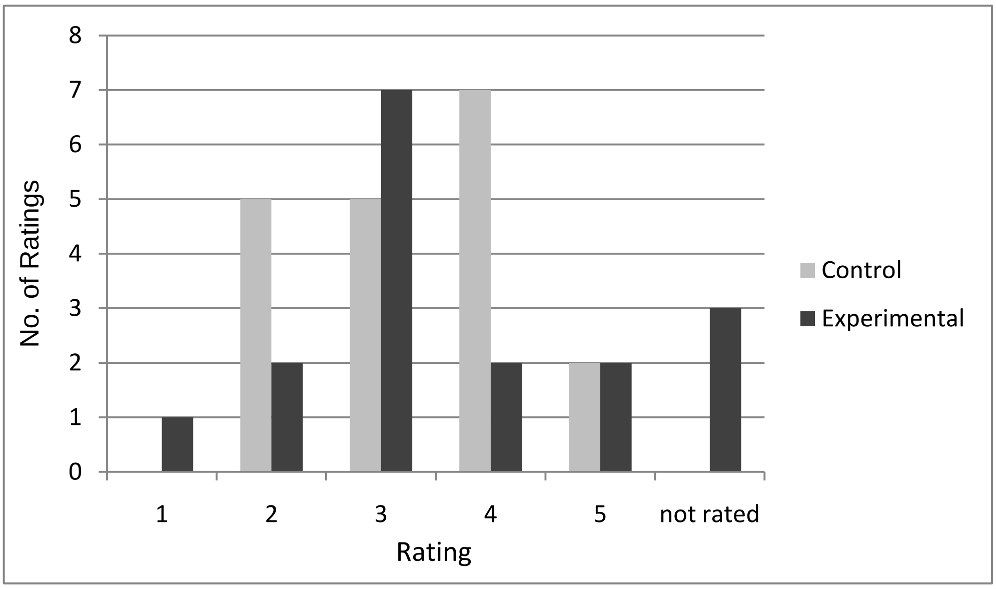


Figure 5.3: Ratings for the enjoyment of using the OSM.

For the enjoyment of using the OSM, the control group got an average rating of 3.22, with similar number of people giving 2, 3 or 4 (shown in Figure 5.3). In the experimental group, most of the students voted 3 and the mean became 3.14. Many comments from students are encouraging. Some said that the OSM gave an approximate view of what they were doing and reminded them what was still needed to do; some found it easy to understand and use; and others commented that the models were useful for building knowledge and helpful with fine details of learning EER modelling. One student complained about being stressed when seeing the red part, which indicated misconception, grew bigger. Another student, who belonged to the experimental group, thought there were too many views to choose from.

After the ratings, students from both groups were asked several yes-no questions. Although the information obtained from these questions was limited, we could see that students had overall positive responses towards the OSM. The first one was whether they examined their progresses to help with identifying weaknesses. Thirteen out of eighteen (about 72.2%) from the control group and ten out of sixteen (62.5%) from the experimental group answered yes. The next question was about whether students thought that being able to examine progresses assisted their learning. Again, around 72.2% control group students answered yes. In the experimental group, twelve out of sixteen (75%) agreed. Students were then asked whether the OSM correctly reflected their ability in solving EER problems. About half of the control group thought it was correct, comparing to 80% of the experimental group who chose yes. In the last question, ten people from the control group and four from the experimental group agreed that they learned EER domain knowledge from the OSM.

The experimental group was given an extra question on which presentation styles they prefer. Nine people voted for skill meters, three liked the concept list, one chose the concept hierarchy, and no one mentioned the concept map. When they were asked for reasons, people who liked skill meters said that it was simpler to understand without effort of thinking; whereas people who preferred concept list and concept hierarchy claimed that these views also had clear representation which was easy to follow, and they contain more details than skill meters.

From the above questionnaire results, we have obtained valuable information that supports our research hypotheses and suggests for future directions in the area. First of all, it is shown again that the OSM, no matter the original or the extended version, has been useful in assisting some students with their learning. Students did show their interest in the OSM and use it during problem solving processes. They were able to understand most parts of the OSM and use it for various purposes, such as locating their weaknesses in the domain, reminding what were left to do, and learning fine details about EER modelling. Secondly, students in the experimental group did show different preferences over the views we provided. Although some students still followed the skill meters, other students were interested in the new views for a variety of reasons. For those who were able to comprehend meanings enclosed in concept hierarchy and concept map, these views had been useful. Therefore, providing them with multiple views in OSM allows their diversity. Last but not least, since the majority of students in the experimental group agreed that the OSM correctly represented their knowledge, these students were likely to improve their meta-cognitive skills once understood the models. They could combine things that they learned from their preferred views to form the knowledge structure of themselves.

We have also learned potential problems from the evaluation study. For example, three students wrote that they did not use some of the new presentations because they were not aware of the existence them. There were a number of people complaining about the complexity of concept map as well. Problems like these, if well treated, can be avoided in the future. We may adapt the current layout. We may also add more information about the OSM. It is actually very helpful to hear these voices from students.

CHAPTER 6

Conclusion

Before we started this research project, we were inspired by the global tendency of using heterogeneous components and comprehensive summaries in user modelling for adaptive educational systems. We began to think about what could be done in our domains and ITSs. Since we had relatively simple formats for presenting student models, what could be delivered to students had been very limited. Therefore, we decided to include more information into the OSM and try to model each student more precisely. The research was aimed to design a multi-view OSM that could allow students' preferences and assist their learning by improving their meta-cognitive skills. An evaluation study was also conducted to measure the effectiveness of the extended version of the OSM.

In the last chapter, we summarize the processes we have taken in this research and draw a conclusion from our findings. Furthermore, we discuss what can be done in the future from where we are standing now. The future of OSM in ITSs does not only include the development of presentations for student models, but also evolve with a number of other issues, which are going to be addressed.

6.1 Research Conclusion

This research has gone through several phases. We started with choosing a suitable system to be the test bed of the research, as the ICTG had developed a number of ITSs.

We needed a domain with reasonable size, so that its knowledge could be represented using a medium number of constraints. Small domains would not need OSM with elaborated details; and large domains would make presenting student model in different formats a very difficult task. We finally decided on EER-Tutor, not only for its domain size, but also for its domain structure, which was well defined and appropriate for a variety of presentations.

Once the choice of tutoring system had been confirmed, we needed to sketch out a list of possible presentation styles for its OSM. An extensive literature research had been done prior to the design phase, and the advantages and disadvantages of well known OSM formats in existing educational systems had been summarized. Taking the characteristics of the domain and the requirements of the system into account, we intended to include the following views in the extended OSM: skill meters, concept list, concept hierarchy, and concept map. Skill meters were previously used on its own to represent student model in EER-Tutor. As one of the simplest presentation styles, it was considered to be easily understandable by most of the students regardless of their learning abilities and knowledge levels in the domain. Concept list was designed as an extended list of skill meters. It would be easily acceptable by students if they were familiar with skill meters. Also, while skill meters provided a high level summary of student model, concept list displayed all the domain topics and the corresponding knowledge levels of a student. Concept hierarchy and concept map further illustrated relationships among domain concepts. The hierarchy delivered a top-down tree view, showing specification and generalization relationships. The map was designed as an EER diagram, which was proposed to not only present complex relationships in the domain but also demonstrate an example of EER modelling.

We implemented these views in EER-Tutor by binding system constraints and domain concepts together. We used a hash table to record the mappings between constraints and concepts. Each concept had been associated with a list of related constraints; and each constraint had been put under at least one relevant concept. These mappings allowed us to calculate how much knowledge had been covered and how much had been correctly learned by a student for each concept displayed in the OSM. We continued using same formulas to calculate these percentages for new views, as we did for skill meters. The layout of the new views was also in line with the appearance of the overall system.

An evaluation study had been conducted within an actual university course. Students were allowed to use the system freely for three weeks and were asked to complete pre-test and post-test. They were also asked to fill out questionnaires at the end of the study. Both objective and subjective data were collected. We calculated the correlations between test scores before and after using the system, together with other indicators, such as the time they had spent on the system and the OSM, as well as how many times they had accessed each view in the OSM. As students were divided into control and experimental groups, we compared their results and concluded that the experimental group had improved significantly whereas the control group had not. In addition, we analyzed students’ answers to questionnaires. We believed that students had preferences over different views of the OSM and they showed interests in the new views we had implemented. It could also be seen that the multi-view OSM helped students to better reflect their own knowledge levels in the domain. These conclusions had supported our research hypotheses.

6.2 Future Work

Even though we have seen encouraging results from the evaluation study, our work on developing student model in ITSs has not ended. After running the evaluation study and considering the valuable feedback from students, we know that there are still lots of things can be done in order to improve our multi-view version of OSM. Besides, having looked at current directions of OSM in existing educational systems, there are many other enhancements we can do in conjunction with multiple presentations in OSM.

Multiple Presentations for OSM in Other ITSs

In this research we have implemented the multi-view OSM in EER-Tutor only. However, there have been a number of ITSs being developed around world. Whether this type of OSM can be widely accepted is still a question. Further research is needed to find out what domains and systems are most appropriate for building such OSM, and vice versa, in what domains and systems such OSM can maximize its effects. For example, in a domain with large amount of information or very loose structures, is implementing concept hierarchy and concept map impossible?

Simplifying Each Presentation in OSM

Including multiple presentations in OSM does not mean making student model complicated. We can include a list of views, but each view should be understandable. From students' comments we learn that the concept map is difficult for most of students even after they have completed the course. Therefore, we may need to think about how to simplify such complex presentations while still trying to deliver as much information as possible. Also, we may be able to add more explanations about such views for students to read.

Relating OSM to Problem Selection

At the moment we are just presenting the OSM to students using different layouts of domain topics and concepts. By reading the percentages on each concept, students can reflect on their learning processes. When students locate where exactly their weaknesses are, they may want to solve problems particularly relating those areas. To allow students select problems themselves, we may map domain concepts not only to constraints but also to problems. If we display the results, for example as a problem list per concept, to students in some good formats, students may be able to practice tasks of certain parts in the domain and hence improve corresponding knowledge levels.

Comparing with Peers in OSM

Another thing may be valuable to do is to allow comparisons among peers in OSM. We may include class average for each concept, or even give the user a ranking. This enables students to be aware of where they are in the group. By finding out their individual positions, they may be able to know whether they have learned enough about a concept at certain time. It is possible that the awareness reminds them about their weaknesses or encourages them to learn.

OSM for Peers and Instructors

There is also trend of developing OSM for peers and instructors [Bull and Hghiem, 2002]. In such OSM, student model can be viewed by other students rather than the owner and human tutors. Again, students can find out their positions by viewing others' models. Teachers can get to know how each student performs in the system and then give directed instructions.

Cross-Course OSM

The last possible future work we want to talk about is cross-course OSM. In the case that a number of ITSs are used in a group of related courses, it may be possible to build an overall student model that summarize student's performance in each tutoring system. We may also include different views from each local system into a global multi-view OSM. By providing such OSM, we may give students opportunity to see their learning processes in different courses as a whole picture. Consequently, it may help student with decision making on long-term learning directions.

References

Anderson, J. R., Corbett, A. T., Koedinger, K. R., and Pelletier, R. (1995). Cognitive Tutors: Lessons Learned. *The Journal of the Learning Sciences,* 4(2):167–209.

Aleven, V., and Koedinger, K. (2000). Limitations of Student Control: Do Students Know When They Need Help? In Gauthier, G., Frasson, C., and VanLehn, K., editors, *Intelligent Tutoring Systems: 5th International Conference, ITS 2000,* pages 292-303. Springer-Verlag GmbH..

Baghaei, N., Mitrovic, A., and Irwin, W. (2005). A Constraint-Based Tutor for Learning Object-Oriented Analysis and Design using UML. *International Conference on Computers in Education ICCE*, pages 11–18.

Baghaei, N., Mitrovic, A., and Irwin, W. (2007). Supporting Collaborative Learning and Problem-Solving in a Constraint-Based CSCL Environment for UML Class Diagrams. *International Journal of CSCL,* 2(2-3): 159-190.

Bloom, B. S. (1984). The 2 Sigma Problem: The Search for Method of Group Instruction as Effective as One-to-One Tutoring. *Educational Researcher,* 13(6):4–16.

Brusilovsky, P., Schwarz, E., and Weber, G. (1996). ELM-ART: An Intelligent Tutoring System on World Wide Web. In Frasson, C., Gauthier, G., and Lesgold, A., editors, *Third International Conference on Intelligent Tutoring Systems, ITS-96,* pages 261-269. Springer-Verlag GmbH.

Brusilovsky, P., and Sosnovsky, S. (2005). Engaging Students to Work with Self-assessment Questions: A Study of Two Approaches. *Proceedings of 10th Annual Conference on Innovation and Technology in Computer Science Education*, pages 251-255, Monte de Caparica. ACM Press.

Bull, S. (2004). Supporting Learning with Open Learner Models. *Proceedings of 4th Hellenic Conference in Information and Communication Technologies in Education*, pages 47-61, Athens.

Bull, S., Cooke, N., and Mabbott, A. (2007). Visual Attention in Open Learner Model Presentations: An Eye-Tracking Investigation. In Conati, C., McCoy, K., and Paliouras, G., editors, *User Modelling 2007: 11th International Conference*, pages 187-196. Springer-Verlag GmbH.

Bull, S., Gardner, P., Ahmad, N., Ting, J., and Clarke, B. (2009). Use and Trust of Simple Independent Open Learner Models to Support Learning within and across Courses. *Proceedings of the 17th International Conference on User Modelling, Adaptation and, Personalization*, pages 42-53. Springer-Verlag GmbH.

Bull, S., and Hghiem, T. (2002). Helping Learners to Understand Themselves with a Learner Model Open to Students, Peers and Instructions. *Proceedings of International Conference on Intelligent Tutoring Systems*, pages 5-13.

Bull, S., and McEvoy, A. T. (2003). An Intelligent Learning Environment with an Open Learner Model for the Desktop PC and Pocket PC. *Artificial Intelligence in Education*, pages 389-391, Amsterdam. IOS Press.

Cimolino, L., Kay, J., and Miller, A. (2003). Incremental Student Modelling and Reflection by Verified Concept-Mapping. *Proceedings of Workshop on Learner Modelling for Reflection, International Conference on Artificial Intelligence in Education*, pages 219-227, Sydney.

Cook, R., and Kay, J. (1994). The Justified User Model: A Viewable, Explained User Model. *Proceedings of the 4th International Conference on User Modeling,* pages 145-150, Hyannis, Cape Cod, USA.

Dimitrova, V. (2003). StyLE-OLM: Interactive Open Learner Modelling. *International Journal of Artificial Intelligence in Education,* 13 (1): 35-78. IOS Press.

Dimitrova, V., Self, J., and Brna, P. (2001). Applying Interactive Open Learner Models to Learning Technicalterminology. In Bauer, M., Gmytrasiewics, P. J., and Vassileva, J., editors, *Proceedings of the 8th International Conference on User Modeling*, pages 148-157. Springer-Verlag GmbH.

Elmasri, R., and Navathe, S. B. (2003). *Fundamentals of Database Systems*. Addison-Wesley, 4th edition.

Freedman, R. (2000). What is an Intelligent Tutoring System? *Intelligence,* 11(3): 15-16.

Gertner, A., and VanLehn, K. (2000). Andes: A Coached Problem Solving Environment for Physics. In Gauthier, G., Frasson, C., and VanLehn, K., editors, *Intelligent Tutoring Systems: 5th International Conference, ITS 2000,* pages 131–142.

Graesser, A. C., Wiemer-Hastings, K., Wiemer-Hastings, P., and Kreuz, R. (1999). AutoTutor: A Simulation of a Human Tutor. *Cognitive Systems Research*, 1(1):35–51.

Hartley, D., and Mitrovic, A. (2002). Supporting Learning by Opening the Student Model. In Cerri, S., Gouarderes, G., and Paraguacu, F., editors, *Proceedings of the 6th International Conference on Intelligent Tutoring Systems, ITS 2002,* pages 453-462, Biarritz, France.

Holt, P., Dubs, S., Jones, M., and Greer, J. (1994). The State of Student Modelling. In Greer, J. E., and McCalla, G. I., editors, *Student Modelling: The Key to Individualized Knowledge-based Instruction*, volume 125 of *Computer Systems and Sciences*, pages 3–35. Springer-Verlag GmbH.

Kay, J. (1997). Learner Know Thyself: Student Models to Give Learner Control and Responsibility. In Halim, Z., Ottomann, T., and Razak, Z., editors, *Proceedings of International Conference on Computers in Education*, pages 17-24.

Koedinger, K. R., and Anderson, J. R. (1993). Reifying Implicit Planning in Geometry: Guidelines for Model-Based Intelligent Tutoring System Design. In Lajoie, S. P., and Derry, S. J., editors, *Computers as Cognitive Tools*, pages 15-45, Hillsdale, New Jersey. Lawrence Erlbaum Associates.

Mabbott, A., and Bull, S. (2004). Alternative Views on Knowledge: Presentation of Open Learner Models. *Proceedings of the 7th International Conference on Intelligent Tutoring Systems*, pages 689-698. Springer-Verlag GmbH.

Mabbott, A., and Bull, S. (2006). Student Preferences for Editing, Persuading and Negotiating the Open Learner Model. *Proceedings of the 8th International Conference on Intelligent Tutoring Systems, ITS 2006*, pages 481-490. Springer-Verlag GmbH.

Mabbott, A., and Bull, S. (2007). Comparing Student-Constructed Open Learner Model Presentations to the Domain. *Artificial Intelligence in Education,* pages 281-288. IOS Press.

Martin, B., and Mitrovic, A. (2002a). Authoring Web-Based Tutoring Systems with WETAS. *Proceedings of the International Conference on Computers in Education*, pages 183–187.

Martin, B. and Mitrovic, A. (2002b). WETAS: A Web-Based Authoring System for Constraint-Based ITS. In de Bra, P., Brusilovsky, P., and Conejo, R., editors, *Proceedings of the 2nd International Conference on Adaptive Hypermedia and Adaptive Web-based System*s, pages 543–546. Springer-Verlag GmbH.

Mayo, M., and Mitrovic, A. (2001). Optimising ITS Behavior with Bayesian Networks and Decision Theory. *International Journal of Artificial Intelligence in Education*.

Mitrovic, A. (1998). Experiences in Implementing Constraint-Based Modeling in SQLTutor. *Proceedings of the 4th International Conference on Intelligent Tutoring Systems*, pages 414–423.

Mitrovic, A. (2002). NORMIT: a Web-enabled Tutor for Database Normalization. In Kinshuk, R., Lewis, K., Akahori, R., Kemp, T., Okamoto, L., Henderson, C., and Lee, H., editors, *Proceedings of the International Conference on Computers in Education*, volume 2, pages 1276–1280.

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

Mitrovic, A. (2005). The Effect of Explaining on Learning: A Case Study with a Data Normalization Tutor. In Looi, C-K., McCalla, G., Bredeweg, B., and Breuker, J., editors, *Proceedings of Artificial Intelligence in Education, AIED 2005*, pages 499-506. IOS Press.

Mitrovic, A., and Martin, B. (2007). Evaluating the Effect of Open Student Models on Self-assessment. *International Journal of Artificial Intelligence in Education,* 17(2): 121-144.

Mitrovic, A., Mayo, M., Suraweera, P., and Martin, B. (2001). Constraint-based Tutors: a Success Story. In Monostori, L., V ́ancza, J., and Ali, M., editors, *Proceedings of the 14th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems*, volume 2070, pages 931–940, Berlin. Springer-Verlag GmbH.

Mitrovic, A., and Ohlsson, S. (1999). Evaluation of a Constraint-based Tutor for a Database Language. *International Journal of Artificial Intelligence in Education*, 10:238–256.

Mohanarajah, S., Kemp, R. H., and Kemp, E. (2005). Opening a Fuzzy Learner Model. *Proceedings of International Conference on Artificial Intelligence in Education*, pages 62-71, Amsterdam.

Ohlsson, S. (1994). Constraint-Based Student Modeling. *Student Modeling: the Key to Individualized Knowledge-based Instruction*, pages 167–189, Berlin. Springerr-Verlag GmbH.

Ohlsson, S. (1996). Learning from Performance Errors. *Psychological Review*, 103(2): 241–62.

Ohlsson, S., and Rees, E. (1991). The Function of Conceptual Understanding in the Learning of Arithmetic Procedures. *Cognition & Instruction*, 8: 103-179.

Perez-Martin, D., Alfonseca, E., Rodriguez, P., and Pascual-Nieto, I. (2007). Automatic Generation of Students Conceptual Models from Answers in Plain Text. Proceedings of the 11th *International Conference on User Modelling*, pages 329-333. Springer-Verlag GmbH.

Polson, M. C., and Richardson, J. J. (1988). *Foundations of Intelligent Tutoring Systems*, Hillsdale. Lawrence Erlbaum Associates.

Psotka, J., Massey, D. L., and Mutter, S. A. (1988). *Intelligent Tutoring Systems: Lessons Learned*, Hillsdale, New Jersey. Lawrence Erlbaum Associates.

Rueda, U., Larranga, M., Arruarte, A., and Elorriaga, J. A. (2006). DynMap+: A Concept Mapping Approach to Visualize Group Student Models. *Innovative Approaches for Learning and Knowledge Sharing, First European Conference on Technology Enhanced Learning*, pages 383-397. Springer-Verlag GmbH.

Suraweera, P., and Mitrovic, A. (2002). KERMIT: A Constraint-Based Tutor for Database Modelling. In Cerri, S. A., Gouarderes, G., and Paraguacu, F., editors, *Proceedings of the 6th International Conference on Intelligent Tutoring Systems*, pages 376–387, Biarritz, France. Springer-Verlag GmbH.

Suraweera, P., and Mitrovic, A. (2004). An Intelligent Tutoring System for Entity Relationship Modelling. *International Journal of Artificial Intelligence in Education*, 14(3):375–417.

Swanson, H. L. (1990). Influence of Metacognitive Knowledge and Aptitude on Problem Solving. *Journal of Educational Psychology*, 82: 306-314.

Weber, G., and Brusilovsky, P. (2001). ELM-ART: An Adaptive Versatile System for Web-based Instruction. *International Journal of Artificial Intelligence in Education*, 12(4): 351-384.

Zakharov, K., Mitrovic, A., and Ohlsson, S. (2005). Feedback Micro-engineering in EER-Tutor. In Looi, C.-K., McCalla, G., Bredeweg, B., and Breuker, J., editors, *Proceedings of the 12th International Conference on Artificial Intelligence in Education*, volume 125 of Frontiers in Artificial Intelligence and Applications, pages 718–725, Amsterdam. IOS Press.

Zapata-Rivera, J-D., and Greer, J. E. (2001). Externalising Learner Modelling Representations. *Proceedings of Workshop on External Representations of AIED: Multiple Forms and Multiple Roles, International Conference on Artificial Intelligence, AIED 2001*, pages 71-76.

Zapata-Rivera, J-D., and Greer, J. E. (2004). Interacting with Inspectable Bayesian Models. *International Journal of Artificial Intelligence in Education*, 14: 127-163.

APPENDIX A

Pre- and Post- Tests

The following is a copy of the pre-test and post-test that was used in the evaluation study described in Chapter 5.

Pre-test (online)

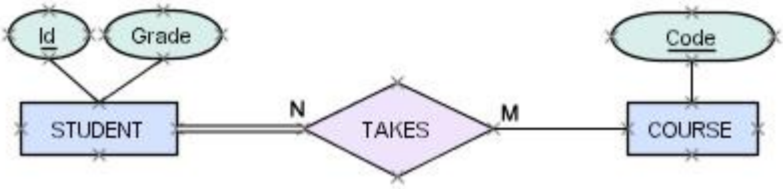
The following test will help you assess your knowledge of enhanced entity-relationship modeling before you begin using EER-Tutor. Please read the questions carefully and select the appropriate answers.

After you submit your solution you'll be able to see correct answers and a detailed explanation of the solutions.

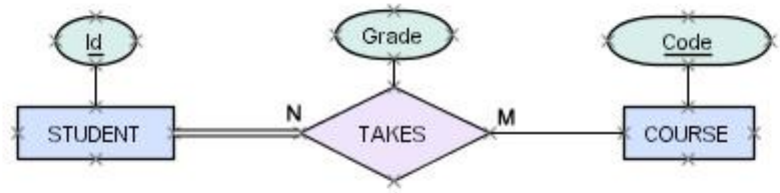
Question 1

Which of the given ER diagrams corresponds best to the following requirements?

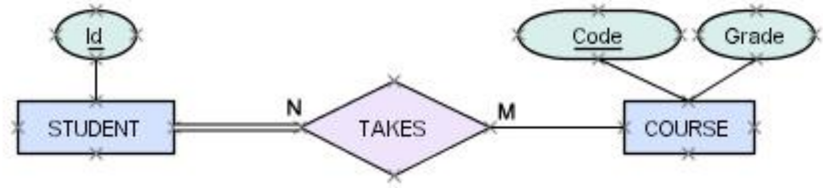
For each course a student has taken, we need to know the final grade. Each course has a unique course code and a student has his/her student id.



a.



b.

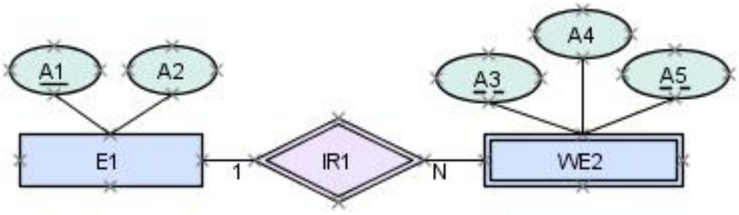


c.

d. Don’t know

Question 2

Is the following diagram correct?



a. Yes.
b. No.
c. Don’t know.

Question 3

The values of the completeness constraint are:

a. Disjoint or partial.
b. Disjoint or overlapping.
c. Partial or total.
d. Overlapping or partial.
e. Don’t know

Question 4

If an entity type has a multi-valued attribute, then

a. Each entity of this type can have one of several values for that attribute.
b. There are some entities of this type that have more that one value for that attribute.
c. Each entity of this type has more than one value for that attribute.
d. There are many valid values for that attribute.
e. Don’t know.

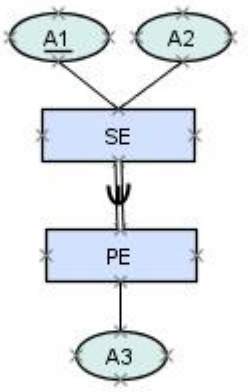
Question 5

A weak entity type participates in the identifying relationship type

a. Always totally.
b. Always partially.
c. Either totally or partially.
d. Don’t know

Question 6

Is the following EER diagram correct?

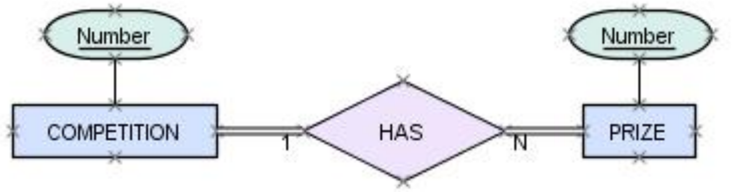


a. Yes.
b. No.
c. Don’t know.

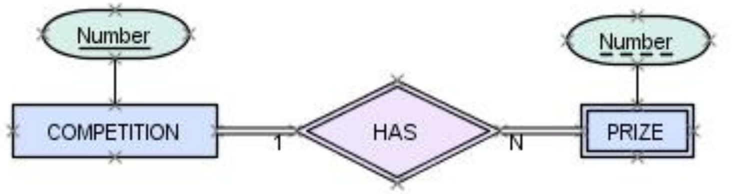
Question 7

Which of the given ER diagrams corresponds best to the following requirements?

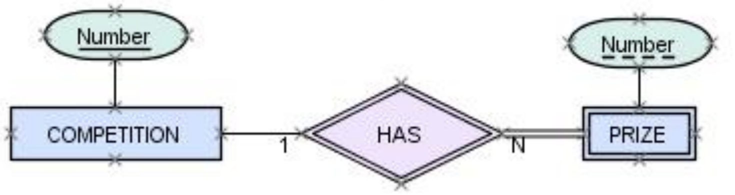
Each competition has a unique number. For every competition there is also a list of prizes identified by numbers unique only within a given list.



a.



b.



c.

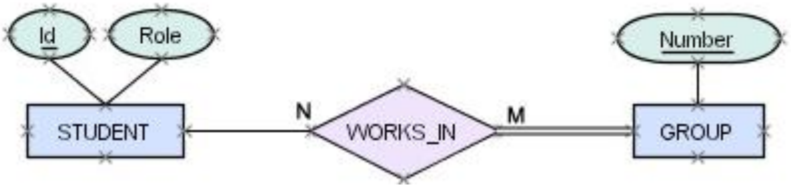
d. Don’t know

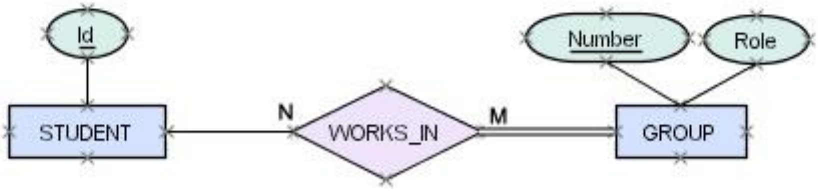
Post-test (in class) User code: _______

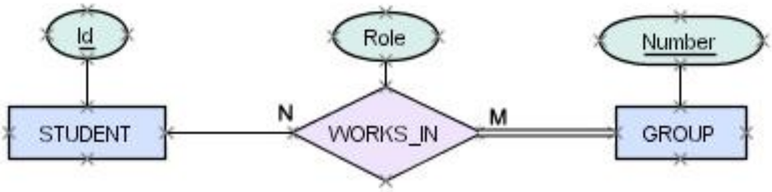
Question 1

Which of the given ER diagrams corresponds best to the following requirements?

Sometimes students work in groups. A student may be a member of several groups, but he/she may have different roles in different groups. Each student has an id, and each group has a unique number.

a. 

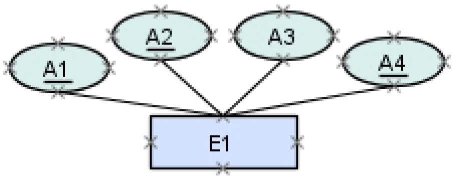
b. 

c. 

d. Don’t know.

Question 2

Is the following diagram correct?



a. Yes.
b. No.
c. Don’t know.

Question 3

A disjoint specialization means that

a. Every entity belonging to the superclass must belong to exactly one subclass.
b. Every entity from the superclass may belong to one or more subclasses.
c. Some entities from the superclass will appear at the level of subclasses.
d. An entity from the superclass may belong to at most one subclass.
e. Don’t know.

Question 4

A derived attribute is

a. An attribute whose values do not exist for every entity.
b. An attribute that has several components.
c. An attribute whose values are optional.
d. An attribute whose values can be derived from other attributes/relationships.
e. Don’t know.

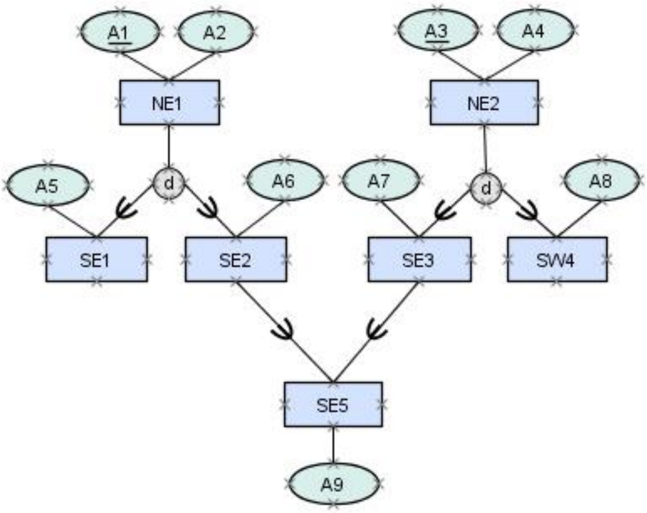
Question 5

A weak entity type can have

a. Exactly one owner.
b. One or more owners.
c. Don’t know.

Question 6

Is the following diagram correct?

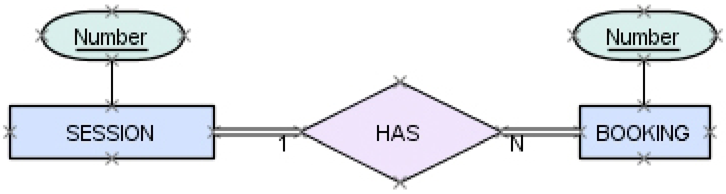


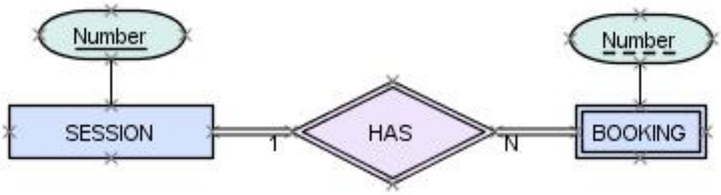
a. Yes.
b. No.
c. Don’t know.

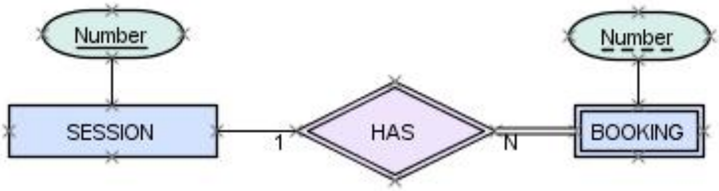
Question 7

Which of the given ER diagrams corresponds best to the following requirements?

Sessions are identified by unique numbers. For some sessions, there might be lists of bookings identified by numbers unique only within a given list.

a.


b.


c.


d. Don’t know.

APPENDIX B

Questionnaires

The following is a copy of the questionnaires used for the control group and the experimental group respectively, in the evaluation study described in Chapter 5.

Questionnaire (control)

Open Student Model in EER-Tutor

Thank you for using EER-Tutor and its student model. Your feedback will be crucial for further improvements of the system and we would be most grateful if you could take time to fill this questionnaire. The questionnaire is anonymous, and you will not be identified as an informant. You may at any time withdraw your participation, including withdrawal of any information you have provided. By completing this questionnaire, however, it will be understood that you have consented to participate in the research and that you consent to publication of the results of the research with the understanding that anonymity will be preserved.

Please specify your university user code here: __________

1. Did you view the student model (the summary of your learning progress) in EER-Tutor? (please circle one)
   a) Never
   b) Only when the system automatically showed the student model after I logged into EER-Tutor as a returning user
   c) I checked the student model myself

If you answered a) in Q1, please do Q2, otherwise please go to Q3:

2. Why didn't you use the student model? (please circle one)
   a) I didn't use EER-Tutor
   b) I didn't know the existence of student model
   c) I knew there was student model but I didn't want to use it

   If you answered c) in Q2, please specify your reason(s):

3. Was it difficult for you to understand the student model in EER-Tutor? (please circle one)

Not at all				Very much
1	2	3	4	5

Why was/wasn't the student model in EER-Tutor difficult? (please comment)

4. Was the student model in EER-Tutor useful? (please circle one)

Not at all				Very much
1	2	3	4	5

Why was/wasn’t the student model in EER-Tutor useful? (please comment)

5. Did you enjoy using the student model in EER-Tutor? (please circle one)

Not at all				Very much
1	2	3	4	5

Why did/didn’t you enjoy using the student model in EER-Tutor? (please comment)

6. Did you examine your progress to help you identify any weaknesses in your EER knowledge? (please circle one)
   a) Yes
   b) No
7. Do you feel that being able to examine your progress assisted your learning? (please circle one)
   a) Yes
   b) No
8. Do you feel the student model in EER-Tutor correctly reflected your current ability to solve EER modeling problems? (please circle one)
   a) Yes
   b) No
9. Did you learn EER domain knowledge from the student model?
   a) Yes
   b) No
10. Do you have any suggestions for improving the student model in EET-Tutor?

Thank you very much for your cooperation!

Questionnaire (experimental)

Multiple Presentations of Open Student Model in EER-Tutor

Thank you for using EER-Tutor and its student model. Your feedback will be crucial for further improvements of the system and we would be most grateful if you could take time to fill this questionnaire. The questionnaire is anonymous, and you will not be identified as an informant. You may at any time withdraw your participation, including withdrawal of any information you have provided. By completing this questionnaire, however, it will be understood that you have consented to participate in the research and that you consent to publication of the results of the research with the understanding that anonymity will be preserved.

Please specify your university user code here: __________

11. Did you view the student model (the summary of your learning progress) in EER-Tutor? (please circle one)

    a) Never

    b) Only when the system automatically showed the student model after I logged into EER-Tutor as a returning user

    c) I checked the student model myself

If you answered a) in Q1, please do Q2, otherwise please go to Q3:

12. Why didn’t you use the student model? (please circle one)

    a) I didn’t use EER-Tutor

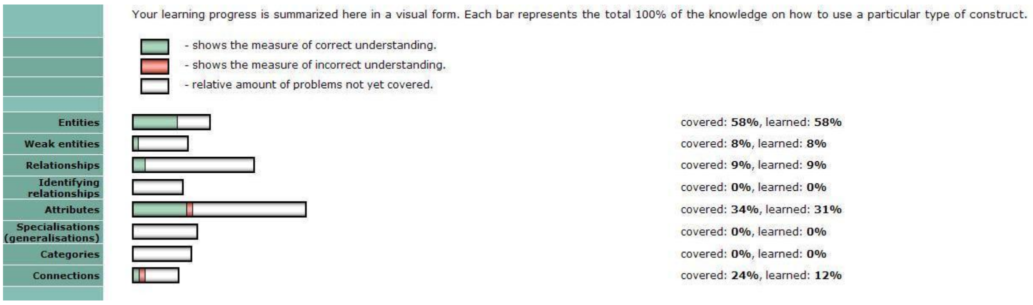
    b) I didn’t know the existence of student model

    c) I knew there was student model but I didn’t want to use it

    If you answered c) in Q2, please specify your reason(s):

13. Which presentation(s) of student model have you examined? (please circle)

    a) Skill-meters (major concepts only)



Your learning progress is summarized here in a visual form. Each bar represents the total 100% of the knowledge on how to use a particular type of construct.

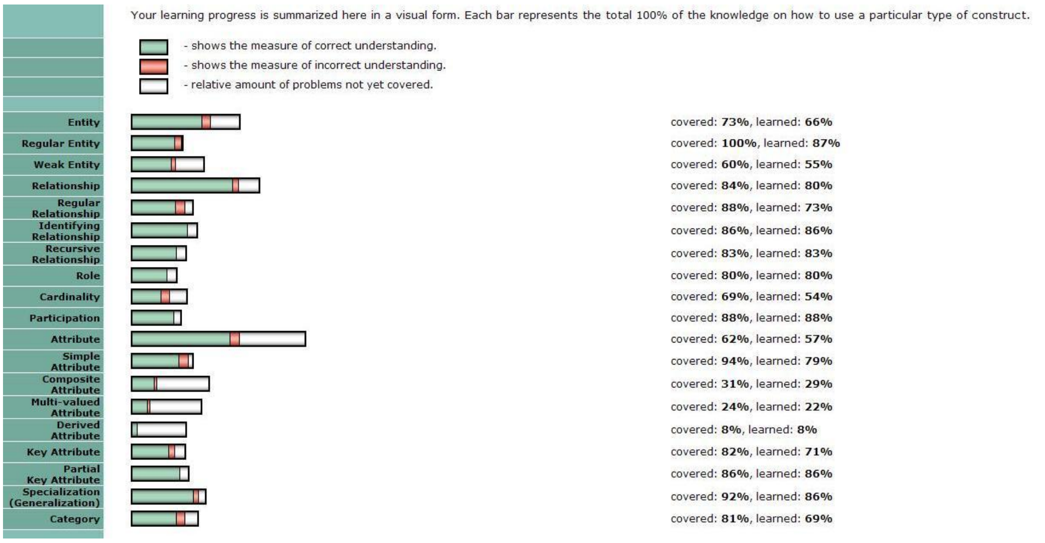
🟩 - shows the measure of correct understanding.

🟥 - shows the measure of incorrect understanding.

□ - relative amount of problems not yet covered.

Entities	covered: **58%**, learned: **58%**
Weak entities	covered: **8%**, learned: **8%**
Relationships	covered: **9%**, learned: **9%**
Identifying relationships	covered: **0%**, learned: **0%**
Attributes	covered: **34%**, learned: **31%**
Specialisations (generalisations)	covered: **0%**, learned: **0%**
Categories	covered: **0%**, learned: **0%**
Connections	covered: **24%**, learned: **12%**

b) Concept list (full list)



Your learning progress is summarized here in a visual form. Each bar represents the total 100% of the knowledge on how to use a particular type of construct.

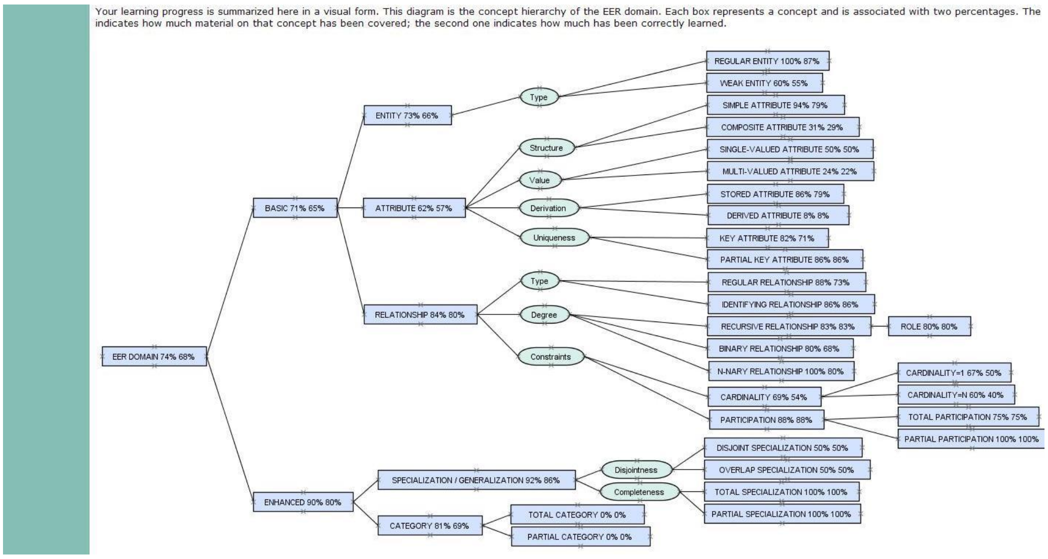
🟩 - shows the measure of correct understanding.

🟥 - shows the measure of incorrect understanding.

□ - relative amount of problems not yet covered.

Entity	covered: **73%**, learned: **66%**
Regular Entity	covered: **100%**, learned: **87%**
Weak Entity	covered: **60%**, learned: **55%**
Relationship	covered: **84%**, learned: **80%**
Regular Relationship	covered: **88%**, learned: **73%**
Identifying Relationship	covered: **86%**, learned: **86%**
Recursive Relationship	covered: **83%**, learned: **83%**
Role	covered: **80%**, learned: **80%**
Cardinality	covered: **69%**, learned: **54%**
Participation	covered: **88%**, learned: **88%**
Attribute	covered: **62%**, learned: **57%**
Simple Attribute	covered: **94%**, learned: **79%**
Composite Attribute	covered: **31%**, learned: **29%**
Multi-valued Attribute	covered: **24%**, learned: **22%**
Derived Attribute	covered: **8%**, learned: **8%**
Key Attribute	covered: **82%**, learned: **71%**
Partial Key Attribute	covered: **86%**, learned: **86%**
Specialization (Generalization)	covered: **92%**, learned: **86%**
Category	covered: **81%**, learned: **69%**

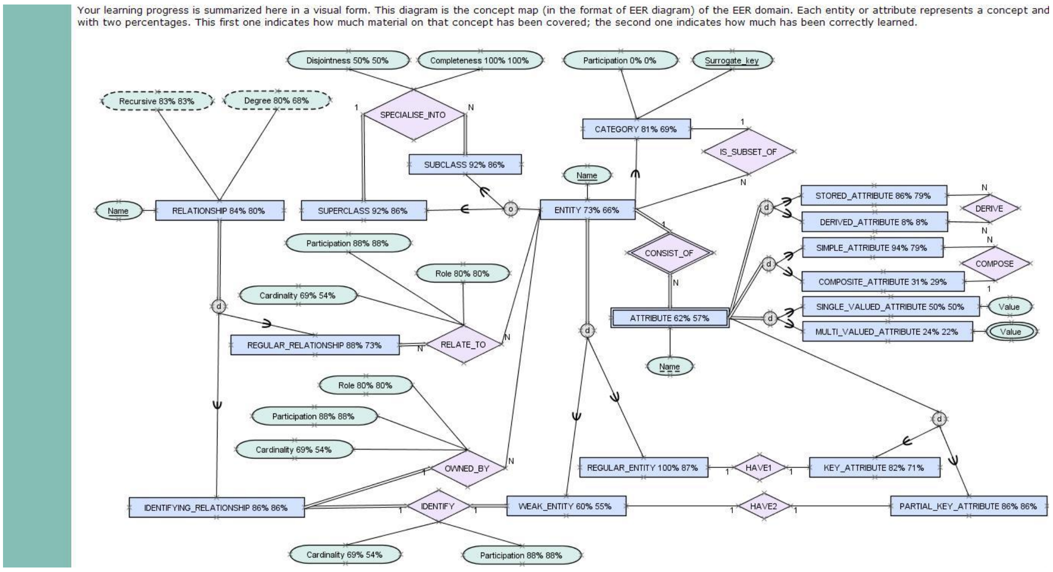
c) Concept hierarchy



Your learning progress is summarized here in a visual form. This diagram is the concept hierarchy of the EER domain. Each box represents a concept and is associated with two percentages. The indicates how much material on that concept has been covered; the second one indicates how much has been correctly learned.

EER DOMAIN 74% 68%
BASIC 71% 65%
ENHANCED 90% 80%
ENTITY 73% 66%
ATTRIBUTE 62% 57%
RELATIONSHIP 84% 80%
Type
REGULAR ENTITY 100% 87%
WEAK ENTITY 60% 55%
Structure
Value
Derivation
Uniqueness
SIMPLE ATTRIBUTE 94% 79%
COMPOSITE ATTRIBUTE 31% 29%
SINGLE-VALUED ATTRIBUTE 50% 50%
MULTI-VALUED ATTRIBUTE 24% 22%
STORED ATTRIBUTE 86% 79%
DERIVED ATTRIBUTE 8% 8%
KEY ATTRIBUTE 82% 71%
PARTIAL KEY ATTRIBUTE 86% 86%
Type
Degree
Constraints
REGULAR RELATIONSHIP 88% 73%
IDENTIFYING RELATIONSHIP 86% 86%
RECURSIVE RELATIONSHIP 83% 83%
ROLE 80% 80%
BINARY RELATIONSHIP 80% 68%
N-NARY RELATIONSHIP 100% 80%
CARDINALITY 69% 54%
CARDINALITY=1 67% 50%
CARDINALITY=N 60% 40%
PARTICIPATION 88% 88%
TOTAL PARTICIPATION 75% 75%
PARTIAL PARTICIPATION 100% 100%
SPECIALIZATION / GENERALIZATION 92% 86%
Disjointness
Completeness
DISJOINT SPECIALIZATION 50% 50%
OVERLAP SPECIALIZATION 50% 50%
TOTAL SPECIALIZATION 100% 100%
PARTIAL SPECIALIZATION 100% 100%
CATEGORY 81% 69%
TOTAL CATEGORY 0% 0%
PARTIAL CATEGORY 0% 0%

d) Concept map



Your learning progress is summarized here in a visual form. This diagram is the concept map (in the format of EER diagram) of the EER domain. Each entity or attribute represents a concept and with two percentages. This first one indicates how much material on that concept has been covered; the second one indicates how much has been correctly learned.

Why did/didn’t you examine these presentations of student model? (please comment)

14. Was it difficult for you to understand the student model in EER-Tutor? (please circle one)

Not at all				Very much
1	2	3	4	5

Which presentation(s) were difficult to understand? (please circle)

a) Skill-meters
b) Concept list
c) Concept hierarchy
d) Concept map

Why was/wasn’t the student model in EER-Tutor difficult? (please comment)

15. Was the student model in EER-Tutor useful? (please circle one)

Not at all				Very much
1	2	3	4	5

Which presentation(s) were useful? (please circle)

a) Skill-meters
b) Concept list
c) Concept hierarchy
d) Concept map

Why was/wasn’t the student model in EER-Tutor useful? (please comment)

16. Did you enjoy using the student model in EER-Tutor? (please circle one)

Not at all				Very much
1	2	3	4	5

Why did/didn’t you enjoy using the student model in EER-Tutor? (please comment)

17. Did you examine your progress to help you identify any weaknesses in your EER knowledge? (please circle one)
    a) Yes
    b) No
18. Do you feel that being able to examine your progress assisted your learning? (please circle one)
    a) Yes
    b) No
19. Do you feel the student model in EER-Tutor correctly reflected your current ability to solve EER modeling problems? (please circle one)
    a) Yes
    b) No
20. Did you learn EER domain knowledge from the student model?
    a) Yes
    b) No
21. Overall, which presentation(s) of student model do you prefer? (please circle)
    a) Skill-meters
    b) Concept list
    c) Concept hierarchy
    d) Concept map

Why do you prefer these particular presentation(s) of student model? (please comment)

22. Do you have any suggestions for improving the student model in EET-Tutor?

Thank you very much!