A Learning System For Entity Relationship Modeling

Mustafa I. Eid
Department of Accounting and MIS, College of Industrial Management, King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia, meid@kfupm.edu.sa

Follow this and additional works at: http://aisel.aisnet.org/pacis2012

Recommended Citation
http://aisel.aisnet.org/pacis2012/152

This material is brought to you by the Pacific Asia Conference on Information Systems (PACIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in PACIS 2012 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.
A LEARNING SYSTEM FOR ENTITY RELATIONSHIP MODELING

Mustafa I. Eid, Department of Accounting and MIS, College of Industrial Management, King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia, meid@kfupm.edu.sa

Abstract

Entity Relationship (ER) Diagram models are commonly used for conceptual data modeling during database design. Developing quality ER data models is a difficult task for both learners and junior modelers. The main reason for this difficulty is that the ER modeling task is cognitively complex. This research proposes a computer-based learning system designed to not only reduce cognitive load in the processes of trial and error, visualization of details, and abstraction but also supports some basic rules of expertise used in ER modeling. The proposed interactive learning system is based on a framework which provides systematic guidance and knowledge support during ER modeling. The framework is based on the use of graphical organizers and animation techniques to simplify problem domain complexity and support visualization during ER modeling respectively. Based on an evaluation of the learning system’s effectiveness and usability, it proved to be better tool for learning ER modeling than the traditional manual method conducted in the classroom. The proposed learning system provides students with the needed hands-on experience in conceptual ER data modeling, thereby speeding up the building of expertise in ER modeling.

Keywords: learning strategies, entity relationship modeling, interactive learning environments, database design
1 INTRODUCTION

The entity relationship (ER) modeling process is based on the ER conceptual data modeling technique first developed by Chen (1976). It represents, in a graphical format, the data needs of the organization’s business operations. While the drawing notations provided by Chen's technique aim to structure/model the problem domain data requirements, the transition step from textual problem domain description into an ERD model using the notations is mainly done based on using heuristics which are neither defined nor structured. In addition to the subject matter knowledge of the problem domain, the transition step requires problem solving skills that involve several cognitive processes. These include trial and error, visualization of details and abstraction, and some basic rules of expertise such as determining the correct class of relationships, resolving M-M (many to many) relationships and correctly determining optional and mandatory semantics. The application of these cognitive processes during ER modeling is somewhat complicated, and thus results in a cognitive overload on the part of learners or novices. This explains why both the production of quality ER models, by learners, and the teaching of ER modeling to new students are challenging. Therefore, to improve student learning or beginner performance in ER modeling some effective learning strategies and/or tools are needed. Existing literature of computer-based learning environments supporting cognitive behavior during ER modeling conveys a lack of research in this area. Thus, the main objective of this research is to develop a computer-based learning system which is capable of achieving the following:

1. Providing a framework which leads to a more effective learning process of ER modeling compared to the traditional method, and
2. Improving effectiveness of learners in performing the ER modeling task. That is, helping learners to produce more accurate and complete ER diagram models.

This research proposes a computer-based learning environment which is designed to support and simplify to some extent the thinking processes during ER modeling. It also offers a process which provides guidance and knowledge support during ER modeling which is both systematic and natural. It is aimed to improve student learning of ER modeling concepts and provides the learner with an opportunity to gain practical skills by taking control of their learning themselves. The graphical organizer approach along with theories of cognitive load, and constructivism represent the theoretical foundation based on which the proposed graphical user interface of the learning system was developed. Based on an evaluation of the learning system’s effectiveness and usability, it proved to be a tool that enhances significantly the learning of ER modeling technique. The proposed learning system provides students with the needed hands-on experience in conceptual ER data modeling, thereby speeding up the building of expertise in ER modeling. After its development the proposed system has been used in the class room by the author to teach ER modeling in systems analysis and design course at university level. It can also be used by businesses in practice to train junior systems developers. Therefore, the research findings will be of great value to both academia and practitioners.

The paper is divided into six sections. Section two, next, discusses the theoretical background underlying the proposed learning system. Section three introduces the proposed system, followed by section four which outlines the research method and presents the proposed system’s evaluation. Section five discusses the evaluation results, and lastly, section six concludes the paper and provides recommendations for future research.

2 LITERATURE REVIEW
2.1 Conceptual ER modeling

Conceptual ER data modeling is an activity undertaken during the systems development process to build a representation of selected semantics about some real-world problem or an application domain (Weber 2003). ER modeling is a problem solving task, and is considered a complex task that depends on critical thinking (Ryan 2000; Shanks 1999). It is usually performed by the application of heuristics and the rules of the used notations. The ER modeling technique proposed by Chen (1976) provides a set of structured notations for representing user data requirements; it, however, does not specifically provide rich and well defined heuristics “know how” to provide support for the cognitive processes/activities involved in formulation of the ER model constructs and semantics.

In general, ER modeling can be viewed as a process which requires five basic cognitive activities. The first activity a modeler will do is to describe the problem domain scenario based on the given user requirements or based on a given interview transcripts of an IS application under investigation. Second, the modeler will need to interpret the given detailed requirements of the problem domain description in order to define its business rules. Business rules describe in a more formal way the business requirements such as transactions, operations, natural relationships between business entities, etc. Here the modeler will try to extract/define business rules that reflect the reality of the user data requirements, thereby avoiding modeling based on literal translation from the sentences of the problem domain description, a mistake commonly made by novices (Batra and Antony 1994). Third, for each business rule, the modeler starts identifying the ER model entities and then linking these entities using relationships (with semantics such as connectivity and cardinality). Fourth, entity attributes are identified including keys. Fifth, these modeling activities run in an incremental and iterative manner in which an ER model segment that represents data requirements in a business rule is added and verified. In each iteration, visualization and verification thinking activities are performed to check correctness and completeness of the ER model against user requirements until a complete and satisfactory ER model is reached. However, it is possible that attributes of an identified entity are defined during the third step before linking the entity with other related entities.

The activity of identifying entities, relationships, and attributes requires the heuristics of abstraction (categorization/classification, generalization, specialization, and aggregation of concepts in terms of constructs such as entities and relationships), as confirmed by (Batra & Davis 1992). This categorization of constructs leads to a simplification of the problem (Srinivasan & Te'eni 1995). These activities are usually followed by a verification of the resulting ERD model against the associated business rules and user views defined in the user requirements. The verification involves visualization of connectivity between instances of related entities. The trial and error process is applied during both the entity abstraction and identification of relationship class (1-M, M-M, or M-M) between proposed entities and relationships, in order to reach the correct ERD model segment for each of the business rules. Thus, the occurrence of these cognitive activities in the manner above mentioned, are challenging for the learner or novice modeler as they result in greater cognitive load/complexity. In addition, the teaching of students to appropriately performing these cognitive activities represent great challenges for IS instructors.

According to Batra et al. (1990) novices generally make lesser mistakes during the identification of entities than the identification of relationships phase. The most common errors are: the incorrect identification of relationships; the incorrect determination of connectivity (in terms of 1-1, 1-M, and M-M); optional and mandatory semantics, and the incorrect determination of cardinality semantics (minimum and maximum connectivity between instances of related entities). The learning of skills and knowledge necessary for the practice of the above mentioned five modeling activities represents a cognitive overload on the part of the learner. Also, the teaching of the conceptual ER modeling represents a challenging task on the part of instructors. Cognitive overload is thinking activities involved in doing a complex task. Cognitive load theory adopts techniques to minimize working memory load in order to accommodate changes in long term memory associated with schema acquisition of learning.
materials/instructions (Sweller 1988). Sweller believes that instructional design should keep cognitive load of learners at a minimum during the learning process. Therefore, to improve student learning, some effective learning strategies/tools are needed to reduce cognitive overload in ER modeling. The use of learning strategies such as examples, and past experience reduces the need for memorization of task rules and allows knowledge building from induction thereby improving the individual's innate ability to perform the task (Ausubel 1963). This view is supported by the research of (Beishuizen et al. 2002). However, Batra (2007) contends what is needed to improve learning of conceptual data modeling is techniques and tools to manage the cognitive complexity of the task. Furthermore, the learning of new knowledge and skills is best achieved following the constructivism approach of learning, specifically practice solving/modeling different problems or cases (Baviskar, Hartle, & Whitney 2009; Wright & Grenier 2009). According to constructivism approach of learning, a learner builds knowledge and skills from experiencing with solving a set of problems and testing their solutions (Dewey 2009).

Graphic representations can be used to understand text and to solve a variety of problems (Gama 2005). The graphical organizer approach is used to support the cognitive processes involved in critical thinking tasks (Hall & Strangman 2007). The graphical organizer approach can provide a framework which assists students in organizing the given problem/application domain information in a format that provides focus and guidance during ER modeling activities. Bock and Yager (2005) for example proposed a paper-based and simple worksheet, which can be considered as a kind of simple graphical organizer presented in a tabular format. The proposed worksheet is aimed at improving student learning of ER modeling. Basically, the function of the worksheet model is to manually organize the problem domain entities and attributes in a simple tabular format to guide the learner’s thinking process during the identification of entities and their attributes at a higher level.

2.2 Computer support for conceptual ER modeling

Current Computer-aided software engineering (CASE) tools provide only drawing tools that aim to improve modelers' productivity during conceptual modeling. These tools do not provide knowledge support nor do they support learning for conceptual ER or UML modeling as Rob and Conronel (2007) emphasized by saying "in the hands of database novices, CASE tools simply produce impressive-looking but bad designs" (pp. 615). During the 1980s, some knowledge-based tools to support ER modeling and object oriented conceptual modeling were proposed (Reiner 1992; Storey 1993). According to Antony and Batra (2002), the efficacy of these tools is not known as the tools have not been evaluated by empirical testing. A review of a selection of these tools is provided by Noah and Lloyd-Williams (1995). However, recent research has produced tools to support learning/performing conceptual data modeling. Antony and Batra (2007) advocate basing such tools on a cognitive framework of problem solving, error prevention, and/or learning. They note that at least three approaches are adopted by current tools designed to provide knowledge support in conceptual modeling:

1. Build a generic tool: The modeling or design knowledge domain are represented and implemented as rule-based knowledge (Storey 1993).

2. Build a repository of conceptual databases: The modeling or design knowledge domain are represented and implemented based on Case-Based Reasoning (Lo & Choobineh 1998).

3. Build a repository of patterns: Patterns are stored in a knowledge base. These patterns can then be instantiated depending on the application (Purao & Storey 1997; Purao 1998; Fernandez & Yuan 2000). However, exceptions to patterns are fairly common.

Suraweera and Mitrovic (2001) have developed the Intelligent Teaching System (ITS), named KERMIT, to assist student learning of ER modeling. After evaluating the ITS tool, the authors claim that the KERMIT environment is an effective educational tool that could be used to enhance student learning in ER modeling. The proposed ITS system adopts the case-based reasoning approach to represent the ER
modeling knowledge. Students’ skills in ER modeling are improved gradually as they practice performing ER modeling problems or exercises. The ITS tool is based on learning cognitive framework. However, the ITS tool does not allow the learner to learn the ER modeling task and skills by doing the actual steps of modeling like an expert. Also, it does not provide support for visualizing the details of entity instances and relationships (connectivity between instances of related entities) and how abstraction of entities and relationships are made from details. While visualization of details and abstraction from details represent a major cognitive activity in ER modeling, many students lack the ability of performing it correctly and easily.

Antony and Batra (2002) proposed a prototype tool CODASYS (COnceptual modeling tool for DAtabase SYStems), to support novice designers in conceptual data modeling. It is a generic tool as it embeds the ER modeling domain knowledge that can be used to tackle any problem domain. It is not aimed at teaching ER modeling but rather at assisting novice designers in the development of an entity relationship diagram model that can be translated to a normalized relational model representation. Scott, Lavoie and Rambow (2001) proposed the LIDA computer tool to support conceptual object oriented modeling through linguistic analysis for a given problem domain. This tool is generic, and aim to assist analysts during the transition from the textual description of the problem domain to the object class model notation. In conclusion, little research work has been carried out in the area of computer-based learning systems for conceptual modeling, in general, and in the use of graphical organizers and animation to support cognitive processes during conceptual modeling in particular. In the following section, the proposed learning system and its development will be outlined.

3 A LEARNING SYSTEM FOR ER MODELING

The proposed learning system for ER modeling is based on a cognitive framework of problem solving and learning. It uses graphical organizers and animation to reduce cognitive complexity during ER modeling following the cognitive load theory. In addition, the constructivism approach of learning is applied here, as the learner takes more control of his learning through increased practice of modeling different problem domains. Since the task of ER modeling heavily depends on both critical thinking and graphical representation of modeling constructs, in the proposed system, GUI (graphical user interface) forms and diagrams are designed based on the graphical organizer approach. They are designed in a way to simplify problem domain complexity, while providing guidance and focus throughout the modeling process. They not only support trial and error, visualization of details thereby supporting abstraction thinking processes, but also embed basic rules of expertise used in determining the correct class of relationships, resolving M-M relationships and correctly determining optional and mandatory semantics. Thus, cognitive load of the learner will be minimized during ER modeling, and learning will be improved as a result of experience with the ER modeling task. As illustrated below, the proposed learning system is based on a systematic framework that forces the learner to perform the key steps of ER modeling. The proposed learning framework is expected to quickly build the self-efficacy of the learner in ER modeling.

To describe what the system functions are, a generic and systematic process for ER modeling needs to be designed. Based on both personal experience and knowledge of developing and teaching ER diagram models as well as the discussion on conceptual ER modeling in Section 2.1, the ER modeling process is defined to include the following steps:

1. Create new problem domain description based on the gathered requirements of an IS application
2. Define business rules of the problem domain
3. Identify entities, relationships, and semantics such as optional and mandatory
4. Resolve M-M relationships, and
5. Define entity attributes.

To achieve a highly interactive learning environment, the forms and help functions are directly accessible from the home page shown in Figure 1. The home page of the system website (Figure 1) provides video
recorded tutorials to demonstrate how the forms that support each of the above five steps function. Figures 2 to 5 depict the forms designed to support steps 3 and 4. The proposed system database plays the role of a data dictionary; it stores previously entered problem domains, and associated business rules along with assumptions and constraints, entities, attributes, relationships, and their semantics. The C# programming language in the MS Visual Studio.net environment was used to build the system. The proposed learning system website has been published online and its URL is found in Appendix A.

![Figure 1. Home page of the proposed learning system for ER modeling](image)

While Step 1 form in Figure 1 allows the learner to enter the description of a problem domain scenario (for example see description of the travel agency problem domain in Appendix A). Step 2 form allows the learner to distill from the problem domain the relevant business rules, there by simplifying the problem domain (for example, a business rule may be: a customer books for many flights, and one flight can be used to book many customers). Step 3 form (in Figures 2 and 3) allows the learner to select and view a business rule and then enter proposed entities. Also, the form in step 3 allows the learner to try choosing a class (one to one 1-1 or one to many 1-M) for one direction, at a time, of the relationship between the proposed entities. Then, in the same form the learner can visualize the connectivity between instances of the proposed entities of the chosen relationship class. This visualization of the proposed entity instances and their connectivity allows the learner to understand the abstraction concept, and validate his or her chosen relationship class. In case the learner found the chosen relationship class is not valid, then he or she can change their choice of the relationship class. Step 4 form (in Figure 4) allows the learner to resolve M-M relationships by providing an easy to use user interface which helps the learner understand the resolution concept. This form is only needed in case of M-M relationships exist. The M-M resolution graphical user interface, in step 4, provides a template which depicts how the bridge entity is introduced between two entities having M-M relationship. This template provides guidelines through the resolution process. It requires the learner to insert a name for the bridge entity. Then, it shows how two new 1-M relationships are introduced between the parent entities and the newly introduced bridge entity (Figure 5). Lastly, the form of step 5 (in Figure 6) allows the learner to define and enter the attributes of an entity and
also specify the key attributes. Step 5 form can also be used to define entity attributes directly after step 3 is performed and before step 4 is done in case of M-M relationship needs to be resolved.

Figure 2. Step 3 Checking Forward Direction of the Relationship

Figure 3. Step 3 Checking Backward Direction of the Relationship
Figure 4. Step 4 Resolution of the M-M relationship

Figure 5. View of the resolved ERD model segment
4 ETHODOLOGY

4.1 Evaluation of the system’s effectiveness

In a bid for proving the value of the proposed learning system for ER modeling, it is necessary to test and evaluate its effectiveness. The evaluation aims to achieve two goals: the first is the verification of the significance of the use of the learning system in relation to enhancing learner performance in ER modeling; and second is to measure the level of satisfaction with the learning system from the perceptive of the learner user. To achieve goal one, the proposed learning system was tested on a group of university level students, known as the experimental group. The evaluation involves an objective comparison between students’ performance in an experimental group who develop an ERD model for a problem domain scenario using the proposed learning system, with students’ performance in a control group who use the traditional ER modeling process to do the same task. Thus, two independent samples/groups were formed from undergraduate students. 104 students participated in the experiment, having similar basic computing background. Participating students are junior level completed one hundred level; introduction to computer applications and second hundred level; principles of information systems courses. They had basic knowledge of database design concepts and ER modeling technique. The experimental (treatment) group had 51 students who were taught ER modeling using the proposed learning system. The control group comprised 53 students who were given in class instructions and laboratory practice based on the traditional method of teaching ER modeling. Both groups were given five times one hour session over a
two weeks period before conducting the experimental assessment task. The traditional teaching instructions given to students include:

1. Carefully read and understand the given problem domain scenario description.
2. Underline nouns that represent things that are candidate entities.
3. Candidate entities should be validated by applying a criteria. The criteria has two questions to be answered by the student: one is to check if the candidate entity has instances or occurrences, second if the user is interested to collect and store data about the candidate entity instances in the desired application database.
4. Define attributes of each of the validated entities.
5. Next, propose relationships between the validated entities based on business rules. Business rules represent the natural association, transactions, or the business operations that requires some entities to be associated with each other.
6. Check the cardinality of 1-1 or 1-M for both directions of the candidate relationships. This checking will result in one of three kinds of cardinality: 1-1, 1-M, and M-M.
7. Resolve the M-M relationships by introducing a bridge entity between the two related parent entities to replace the M-M relationship, and then add 1-M relationship between each of the parent entities and the new bridge entity. Then, define the attributes for the new bridge entity.
8. The above activities run in an iterative manner until all sentences describing the business rules of the given problem domain scenario are considered.

A moderately complex problem domain scenario was prepared (see Appendix A) with the same problem scenario and test instructions, given to the students in both groups. The students in the control group were requested to manually develop the ER diagram model based on their knowledge and skills gained from the traditional teaching of ER modeling. The students in the experimental group, however, used the learning system for ER modeling to develop the ER diagram model. The time allowed for both groups to perform the ER modeling task was uniform as 1.5 hours.

The required ER diagram models in the test involved three activities: identification of entities; relationships; and the resolution of any M-M relationships. These three activities were designed with the aim of evaluating the effectiveness of the proposed learning system with regards to the correct identification of entities and relationships compared with the results from the traditional ER modeling process. Since the students in the experimental group will not perform by themselves the mandatory and optional semantics of the identified relationships, it is not requested from the students in the control group to perform these semantics in their ERD models. Therefore the determination of these semantics was excluded from the experiment. The implication for learning of having the determination of these semantics automated is that the determination of optional and mandatory semantics depends on answering the question that “Must every instance of entity one must be associated with at least one instance of the other related entity two?” (Figures: 2 and 3). If the answer is Yes, then entity two is mandatory to entity one, else entity two is optional to entity one. The resulting optional and mandatory semantics are shown on the built ERD model in Figure 4. Also, the identification of entity attributes is excluded from the experimental task since that students in both groups identify entity attributes in a similar way.

After conducting the experiment the resulting ERD models from both groups were graded by a computer science graduate student without knowing what models belong to what group. A standard marking scheme was used and two sets of percentage grades were generated. Table 1 shows that the average of the control and experimental groups were 50.56 and 65.09 percent respectively. The standard deviations of the control and experimental groups were 10.72 and 9.27 respectively. Since the size of the samples exceeded 30 observations, the resulting grades data could be assumed to follow the normal distribution according to the Central Limit Theorem. This statistical data analysis was to test improvement in student performance by using the learning system. SPSS version 16 was used to implement two tests namely: the Levene's test and the Independent Samples T-test, with the results shown in Tables 1 and 2.
The first test applied was the Levene's test. This test examines the homogeneity of the data in the two samples. The Levene's test evaluates the assumption of equal variances of the two samples. To perform this test, two hypotheses are made: \( H_0: \sigma_1^2 = \sigma_2^2 \) and \( H_1: \sigma_1^2 \neq \sigma_2^2 \). The calculated \( P \) value, generated based on the Levene's test, was 0.131. Since the \( P \) value is greater than the alpha value of .05, then, the null hypothesis cannot be rejected. This indicates that the two samples have equal variances; hence the T-test based on equal sample variances must be used. To perform the T-test, two hypotheses are made, \( H_0: \mu_1 = \mu_2 \) and \( H_1: \mu_1 \neq \mu_2 \). The calculated \( P \) value is .000 which is less than .05, thus the null hypothesis relating to equality must be rejected. This shows that on average student performance in the experimental group was higher than the control group students’. Despite the relatively small difference (around 15 percent) between the two sample means, the result of the T-test showed that the difference between student performances in the two samples is statically significant. In addition, if the students in the experimental group were given more time to use the tool, this difference in performance, with respect to the control group students, is expected to increase.

<table>
<thead>
<tr>
<th>Group</th>
<th>Size N</th>
<th>Mean</th>
<th>Standard Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Control</td>
<td>53</td>
<td>50.57</td>
<td>10.73</td>
</tr>
<tr>
<td>Experimental</td>
<td>51</td>
<td>65.10</td>
<td>9.20</td>
</tr>
</tbody>
</table>

*Table 1. Group Statistics*

<table>
<thead>
<tr>
<th>Test</th>
<th>( P ) value</th>
<th>DF (degrees of freedom)</th>
<th>Confidence level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Levene's Test</td>
<td>0.131</td>
<td></td>
<td>95%</td>
</tr>
<tr>
<td>t-Test (Equal variances assumed)</td>
<td>0.000 (2-tailed)</td>
<td>102</td>
<td>95%</td>
</tr>
</tbody>
</table>

*Table 2. Levene’s Test and Independent Samples T-test*

### 4.2 Usability study survey

A usability questionnaire was developed and distributed to the experimental group. The usability questionnaire aimed to obtain user views on two dimensions of user satisfaction of the proposed system prototype. The two dimensions are: ease of use and system’s usefulness with respect to supporting the ER modeling process. The questionnaire was divided in three parts (Table 3):
1. Five scale questions 1 to 5 to investigate ease of use,
2. Four scale questions 6 to 9 to investigate system usefulness, and
3. Last question is on feedback comments.

The nine questions were adopted from the study by Lund (2001) with slight alterations to account for the setting of the study. Questions 6 to 9 aim to measure the ease of, speed of, and quality of performing ER modeling using the proposed system compared to performing ER modeling manually based on their knowledge of ER modeling. The questionnaires were completed by 51 students in the experimental group. The results in Table 3 show both a positive impression and useful feedback was obtained.

### 5 DISCUSSION OF FINDINGS
The results of the applied independent samples T-test show that significant difference between the mean of the experimental group and the mean of the control group. This indicates that ER modeling performance of the students in the sample using the proposed learning system was enhanced. This indicates that the proposed learning system has been statistically proven to be an effective tool for learning ER modeling. The results, presented in Table 3 of the evaluation of the ease of use (questions 1 to 4) part of the proposed system, show that the majority of the participants either agreed or strongly agreed that the user interface design: (1) is intuitive, (2) easy to navigate, (3) system functions are logically grouped/organized, and (4) it is easy to learn to use the system. With respect to question 5, users’ responses show that just over half (54.2% [51.3% and 2.9%]) of the participating students agreed or strongly agreed that the proposed system GUI is pleasant, 22.9% were neutral, 22.9% disagreed, and 2.9% strongly disagreed. However, large majority of the same participating subjects have agreed and strongly agreed in response to questions 1 to 4 indicating that the proposed system is intuitive, easy to use, and easy to learn. Therefore, the proposed learning system GUI may be considered generally pleasant.

S.D. = Strongly Disagree, D. = Disagree, N. = Neutral, A. = Agree, S.A. = Strongly Agree

Table 3. Usability Questionnaire Results

With respect to the evaluation of the system’s usefulness, the results presented in Table 3 (questions 6-9) show for question 6 a vast majority (91.4% [60.0% and 31.4%]) of students agreed or strongly agreed that the proposed system will make ER modeling easier to perform. Furthermore, the majority of students agreed or strongly agreed in response to questions 7, 8, and 9. In addition, a few comments giving feedback were received from respondents. Most of the comments are focused on adding more functionality such as:

- Producing the complete ER diagram model for the whole problem domain
- Printing output ER diagram models in PDF format.
• Determining the cardinality semantics of minimum and maximum connectivity between entity instances.

According to the findings of the statistical T-test and system usability questionnaire, it can be deduced that the proposed system for teaching ER modeling has achieved the two research objectives of:
1. Helping learners to learn ER modeling easier and faster than the traditional method of learning.
2. Improving learner performance in ER modeling through providing the support and guidance to produce more accurate ER diagram models.

In summary, the added value of the proposed system is three-fold: (1) it simplifies the task of ER modeling for learners by using graphical forms to organize problem details which reduce cognitive complexity or cognitive load, (2) guiding learners to follow a systemic process based on best practices in ER modeling, and (2) it enhances their performance in ER modeling by producing more accurate models. In addition, over the time, the stored ER models developed for previous problem domains in the system database will provide a knowledge base of previously modeled applications which can be shared by learners as cases to learn from. The proposed framework and the guidance and support for cognition it provides for represent a valuable contribution to the body of knowledge of both the pedagogy of conceptual data modeling and the practice of ER modeling.

6 CONCLUSION AND FUTURE RESEARCH

A literature review of conceptual data modeling has found that only little research has been carried out in the area of computer-based learning systems supporting conceptual ER modeling. A framework and a system prototype for learning ER modeling were designed. The forms were designed to simplify problem domain complexity while systematically providing guidance, visualization of details using animation throughout the ER modeling process. The forms also embed some rules of expertise formulated to determine optional and mandatory semantics between entities. Effectiveness and usability of the proposed system prototype were evaluated by implementing an independent sample T-test and usability questionnaire respectively on a sample of university students. The statistical T-test results showed that the proposed prototype enhanced participating students’ learning and performance in ER modeling. The usability questionnaire results showed that the majority of participants agreed or strongly agreed that the proposed prototype is easy to use and useful for learning and performing ER modeling. The idea of using graphical organizers and animation to support cognitive processes used in this research can be applied to other conceptual modeling techniques such as UML Use Case and Activity diagrams, DFD, or business process modeling diagrams. Future research can be focused on further developing the proposed learning system to support learning in both conceptual data and object modeling. Moreover, additional research could be carried out how best the proposed learning system can be integrated with one of the current CASE tools to promote and support learning while drawing ERD models.

References


Acknowledgements

It is imperative to acknowledge and appreciate the support of the DSR at KFUPM, Saudi Arabia, through their research grant (Project# JF060008), which lead to the outcome of this research paper.

Appendix A: Travel Agency Problem Domain

A local travel agency wishes to develop a database application to automate its customer holiday booking activities. The travel agency has a number of branches in Saudi Arabia. Customers book a flight and/or accommodation through a branch. Then, the branch generates an invoice for a holiday booking. The customer will need to pay the invoice. Available accommodation includes: hotel room and an apartment.

Task: Develop an entity relationship diagram model for the above application. Time allowed is 1.5 hours.

1. Determine entities for the above travel agency application. (40 points)
2. Determine 1-1, 1-M and M-M relationship (40 points)
3. Resolve M-M relationships. (20 points)

URL of the learning system: [http://www2.kfupm.edu.sa/erdtool/](http://www2.kfupm.edu.sa/erdtool/)

User name: user
Password: user