1999

Adaptive User Models for the Design of Intelligent User Interfaces

Lisa Martha Hunt
University of North Florida

ADAPTIVE USER MODELS
FOR THE DESIGN OF INTELLIGENT USER INTERFACES

by

Lisa Martha Hunt

A thesis submitted to the
Department of Computer and Information Sciences
in partial fulfillment of the requirements for the
degree of

Master of Science in Computer and Information Sciences

UNIVERSITY OF NORTH FLORIDA
DEPARTMENT OF COMPUTER AND INFORMATION SCIENCES

March, 1999

The thesis "Adaptive User Models for the Design of Intelligent User Interfaces" submitted by Lisa Martha Hunt in partial fulfillment of the requirements for the degree of Master of Science in Computer and Information Sciences has been

Approved by the thesis committee: Date

Signature Deleted

3/29/99

Dr. Krissten N. Cooper
Thesis Advisor and Committee Chairperson

Signature Deleted

3-29-1999

Dr. Kenneth E. Martin

Signature Deleted

3/29/99

Dr. Robert F. Roggio

Accepted for the Department of Computer and Information Sciences:

Signature Deleted

4/13/99

Dr. Judith L. Solano
Chairperson of the Department

Accepted for the College of Computing Sciences and Engineering:

Signature Deleted

4/14/99

Dr. Neal S. Coulter
Dean of the College

Accepted for the University:

Signature Deleted

4/15/99

Dr. William J. Wilson
Dean of Graduate Studies

ACKNOWLEDGEMENT

I wish to express my sincere appreciation and gratitude to my thesis advisor, Dr. Krissten N. Cooper, for her direction and for the extraordinary amount of time she has committed to this research. I also wish to thank her for her encouragement and support towards my research as well as all of my endeavors in the department. I am extremely grateful to her.

I wish to express my sincere appreciation to my thesis committee, Dr. Kenneth E. Martin and Dr. Robert F. Roggio for their time, encouragement and support and for always being willing to listen. I am extremely grateful for the time and advice they have shared with me during the last three years.

Lastly, I wish to express my appreciation to my mother for her incredible love, encouragement and support throughout the last three years and my brother for never failing to make me laugh.

CONTENTS

CONTENTS (Continued)

CONTENTS (Continued)

CONTENTS (Continued)

FIGURES

TABLES

ABSTRACT

The objective of this research is to determine the effects over time of a dynamic system that adapts itself to a user's current state of expertise, in terms of the application domain, by constantly monitoring the user throughout use of the system, placing them in appropriate user models when this expertise has changed.

A dynamic system, named ER-by-Design version 2.0, is presented, consisting of an inference component, a help system, a help/assistance screen, and user models. The user models are responsible for adapting the system interface to the level of expertise of the user. The system monitors and analyzes a user's interactions in order to evaluate user expertise, placing the user in the most appropriate model based on this evaluation.

Through analysis of data collected from participants' sessions with both versions of the system, it is shown that over time, through the use of ER-by-Design version 2.0, users accessed help less often and perceived the

system as more beneficial when compared to a system with a static, generic interface. In addition, users who had the least experience with ER modeling concepts created more correct diagrams with ER-by-Design version 2.0 than with a static version of the system.

Chapter 1

INTRODUCTION

Users vary in terms of levels of knowledge and expertise pertaining to the application domain of the system with which they are interacting. For instance, users working with an application that leads them through the creation of an Entity-Relationship (ER) model, may be at different levels of mastery of concepts of the model. Some users may be new to the ER model and only beginning to grasp the concept of representing basic entities. At the opposite extreme, other users may be quite proficient in their knowledge of the concepts of the ER model and in creating ER diagrams with a high level of complexity. It is important that the system serves users in both cases efficiently and effectively.

A useful technique in achieving different interfaces for different types of users is the user model. Users are classified according to stereotypes that take into consideration the characteristics of each user, such as their level of expertise. The user model allows the system to adapt its interface to the proficiency level of each user, thus serving users in a more efficient manner.

Much research has been done concerning user models, emphasizing their effectiveness in aiding the user by taking into consideration specific characteristics of stereotypes a user may possess, and using those stereotypes to modify the interface of a system to correspond and adapt to the user. Most of these studies focus on placing the user in an appropriate user model upon the first interaction with the system. The users remain in the same model, over time, without the consideration that the characteristics of the user that made the model appropriate in the beginning may have changed over time with use of the system and more proficient knowledge of the application domain. The hypothesis of this research is that over time, a dynamic system that adapts itself to a user's current state of expertise, in terms of the application domain, by constantly monitoring the user throughout use of the system, and placing them in appropriate user models when this expertise has changed, will provide a more effective and efficient environment for users when compared to a system with a static, generic interface based on a static user model.

1.1 Research Goals

To investigate this hypothesis, a system that employs user models and an inference engine is developed. The inference engine monitors the user's interaction with the system in order to place the user in the appropriate user model when it has been demonstrated that user expertise has improved. Experiments are conducted studying the use of this system compared to a system with a static user model utilizing a static interface. Data is collected and statistically analyzed to establish if over time the adaptive system provides a more efficient and effective experience for the user compared to the system with the static interface.

1.2 Overview of Research

The following chapters present research to support this hypothesis. Chapter 2 reviews the literature and background information related to this research. Research concerning intelligent user interfaces, user models and methods of assessing user proficiency is presented.

Chapter 3 presents the design of the system, ER-by-Design version 2.0, intended to test the research's hypothesis. Details of the system, such as the inference component, the help system, the help/assistance screens and the user models are explored. Chapter 4 presents the criteria for testing the hypothesis as well as the analysis of the data and Chapter 5 summarizes results, states conclusions, and presents areas for future research.

Chapter 2

OVERVIEW

This chapter surveys literature related to the research and development of a dynamic system that employs user models based on user expertise of the application domain, and monitors the user in order to place them in the most appropriate user model when user expertise has changed. The focus of this review is on the history and background of intelligent user interfaces, user models and methods for gauging user proficiency.

2.1 Intelligent User Interfaces

Intelligent user interfaces (IUIs) are defined as "human machine-interfaces that aim to improve the efficiency, effectiveness, and naturalness of human-machine interaction by representing, reasoning, and acting on models of the user, domain, task, discourse, and media" [Maybury98, page 2]. They are different from traditional interfaces because they "represent and reason about the user, domain, task, media, and situation" [Maybury98, page 2]. The research in this area focuses on research in the specific domains of human-computer interaction, ergonomics, cognitive science and artificial intelligence [Maybury98].

2.1.1 Importance and Benefits

Computer systems being built today are more complex than their predecessors and are conveying and processing larger amounts of information as well as dealing with more complex task structures, real time performance, and the use of agents [Sullivan91]. For these reasons, it is important for computers to "achieve the ability to reason and make decisions on their own" [Sullivan91, page ix]. In addition, there is an "explosion of available materials" [Maybury98, page 1], creating a "need for more effective, efficient, and natural interfaces to support access to information, applications, and people" [Maybury98, page 1].

The benefits of IUIs are numerous and include user benefits such as adaptability, context sensitivity, task assistance, comprehension of multimodal input, generation of multimodal presentations, automated completion of tasks and management of the interaction, to name a few [Maybury98].

2.1.2 Use of Natural Language and Direct Manipulation

Cohen et al. studied the integration of natural language and direct manipulation to see how it aided the user interface. They found that the use of the two methods together proved to be more useful in helping the user achieve his goal than the use of one technique alone. [Cohen98]

2.1.3 Relationship to Artificial Intelligence

The field of artificial intelligence (AI) has contributed much to the work being done in intelligent user interfaces "including the use of knowledge representations for model-based interface development tools, the application of plan generation and recognition in dialog management, the application of temporal and spatial reasoning to media coordination, the use of user models to tailor interaction, and so on" [Maybury98, page 3]. AI techniques have much to offer. "This belief is founded on a set of techniques that ease the solution of large, complex problems that challenge solution through algorithmic techniques" [Miller91, page 2]. "The hope is that the field can merge the strengths of AI - a broad, powerful set of representational and reasoning techniques for computing about complex domains and tasks - with the strengths of good user interaction techniques - a means of direct user access to these concepts, providing a broad communications channel between the users and the computational engine" [Miller91, page 2].

2.1.4 History

Intelligent interfaces are not new. Much of the oldest work in AI, mostly concerning natural language and problem solving, focused on research on intelligent interfaces. The early work focused on a natural language discourse that was reinforced by the teletype, the current technology at the time. [Miller91]

The focus has shifted in more recent years due to the development of graphical interfaces, its result being that "interfaces need no longer be bound to a linguistic style of interaction" [Miller91, page 3]. It is believed that "graphical interfaces can make it easier for intelligent systems to determine the meaning underlying users' actions: instead of having to search for the meaning in a natural-language statement, a graphical interface can be built around important concepts in the task and domain at hand, making the intent of a users' actions immediately accessible to an underlying reasoning system" [Miller91, page 3].

In the 1990s, advances have been made leading to such commercial applications as e-mail filters and Microsoft's Office Assistant, which uses Bayesian-based user models, as well as the implementation of agents [Maybury98].

2.1.5 Importance of a Good Intelligent User Interface

Intelligent user interfaces should be "learnable, usable and transparent" [Maybury98, page 1]. A good interface should be thought of as a member of a team and "in particular, the member of the team responsible for getting things done on the system" [Sullivan91, page viii]. Intelligent user interfaces should address questions such as how to make the interaction clearer and more efficient, how to offer better support for the user's tasks, plans and goals, and how to present information efficiently [Sullivan91]. IUIs should promise more efficient, effective and natural interaction [Maybury98].

"Intelligent systems only perform as well as their representations of the task they are trying to perform and of the world they are trying to perform it in" [Birnbaum97, page 173]. User interfaces "must be judged by the ease and effectiveness with which they are used by people to perform tasks" [Birnbaum97, page 175].

Birnbaum et al. share the belief that intelligence should only be added to a system if it can be implemented well, else it might impede the user, leading to user frustration. In designing an intelligent interface it is important to weigh the advantages of adding AI to the interface versus the consequences. The consequences of many intelligent interfaces are the likelihood of the interface making mistakes and the cost of these mistakes, as well as the slowness and seemingly unresponsiveness of the interface due to the addition of AI techniques. Usability is an issue as well, as many developers add AI to create a more natural interface without crafting the interface to support the AI, resulting in an interface that is less usable than before. One approach is to remain conservative when applying AI to an interface in order to avoid "the wrath of the user" [Birnbaum97, page 175]. Some techniques for the successful creation of intelligent interfaces are to suggest rather than act, thus not disturbing the user's interaction, to operate in real time, and to watch the user's actions. [Birnbaum97]

2.2 Adapted, Adaptable and Adaptive Interfaces

An adapted user interface is adapted to the end user at design time, an adaptable user interface is one in which the end user may change the characteristics or functionality, and an adaptive user interface changes its characteristics dynamically at run time with regard to the user's behavior [Schlungbaum97].

Miller, Sullivan and Tyler present two approaches concerning these types of intelligent interfaces, the model world approach and the notion of agents. The model world approach "enables the user to communicate directly with the system concerning concepts, goals, and plans; it leaves the system with the responsibility to implement low-level actions necessary to achieve these goals"

[Miller91, page 7]. The interface is not left with the burden of inferring the user's plans and goals from the user's actions. This approach would be used in the creation of adapted and adaptable interfaces. The other approach, the notion of the agent, consists of an inferential component that examines the user's actions and infers the plans and goals of the user from these actions. The appearance of behavior of the interface is then modified accordingly [Miller91].

To some, the agent approach is the only one that exhibits true intelligence, as some researchers define an intelligent interface at its extreme as "an intelligent agent that embodies some of the key capabilities of a human assistant: observing and forming models of the world and the user; inferring user intentions based upon those observations; and formulating plans and taking actions to help the user achieve those intentions" [Tyler91, page 85].

There is much controversy concerning the two approaches and advocates of each approach have their reasons for believing their approach is the best method in the creation of an intelligent user interface. Advocates of the model world approach are skeptical in the use of a tractable method of goal recognition and believe that inference is too difficult to achieve. On the other hand, the advocates of the agent approach believe it is important for the interface to take the initiative and take the position that this initiative can be achieved. Both approaches are reflected in much of the current literature concerning intelligent user interfaces. [Miller91]

2.2.1 Adaptive Systems

CHORIS, the Computer-Human Object-oriented Reasoning Interface, developed by the Intelligent Interfaces Research Group at the Lockheed Artificial Intelligence Center, is an adaptive system that "is designed to enable a wide range of users to interact effectively with varying types of complex applications" [Tyler91, page 85]. It consists of a "set of domain independent reasoning modules driven by domain-specific knowledge bases" [Tyler91, page 85]. The knowledge bases include models of the user, domain and the interface itself. CHORIS also consists of the Plan Manager, that is used to interpret user actions and infer user intentions, as well as the Adaptor that is used to modify the interface features.

CHORIS can be used with several domains, one such domain being an emergency crisis management system. In this type of system the user can view a map of a geographical area and is able to respond when an emergency situation arises. The user can ask CHORIS questions and CHORIS responds to the question as well as displaying pertinent information to the screen in order to aid the user in his task. [Tyler91]

Sukaviriya and Foley present the User Interface Design Environment (UIDE) that uses the knowledge of an application in presenting the application's interface and in presenting automated help. The UIDE includes user models to "evaluate when an interface should adapt, and provide help which is adapted to the user" [Sukaviriya93, page 111]. Their approach is to have the system suggest these adaptations to the user first, allowing the user to always maintain control over the acceptance of the adaptation.

The system keeps a history of interactions in order to decide when adaptation should take place. An example of this concept is when an action has been successfully invoked, leading the system to assume that the user knows about the action. At that time, one count is added to a special slot in the user model to record this action. Similar recordings are made when a user cancels an action or requests help with an action. These records are useful when trying to evaluate the knowledge of a user. [Sukaviriya93]

Meyer, Yakemovie and Harris believe an important part in designing an adaptive interface is determining which aspects of the system will adapt in response to changing conditions [Meyer93]. Some of the ways the system may adapt are: task allocation or partitioning, where the system performs part or all of the task, and interface transformation, where the system changes the content and form of displayed information in order to make completion of the task easier. Other ways in which a system may adapt are adapting functions available to each user and helping the user to adapt by such methods as intelligent tutoring.

Specifying the conditions that will cause the system to adapt is just as important. The system may adapt to certain characteristics of the user, task, domain or environment such as the user experience with the task, previous experience, the user aptitude, preferences and demographics, the task complexity and/or frequency, the probable workload and the physical conditions to name a few. Selecting the data to drive the adaptation can be quite challenging. There are several types of collectable data, one of these being stable user information. This form of data collection is the easiest to collect and consists of information such as job title and education. Other forms of collectable data are workload data, speed data and accuracy data. [Meyer93]

Adaptation can be divided into the three categories: user requested, prompted by the system, or automatic. Benyon and Murray see the first two categories as forms of customization and believe the distinction between customization and automatic adaptation is important [Benyon93]. "Automatic adaptation presents an altogether different challenge, because the computer system needs to contain a detailed and explicit representation of the user (a user model), of itself (a task or domain model) and of the user-system interaction (an interaction model) if it is [to] adapt appropriately" [Benyon93, page 115]. They also believe the level of the system to be adapted as well as the user characteristics need to be considered and that the system can adapt at the levels of description represented in the domain model or in the user model [Benyon93].

2.3 Modeling the User

Intelligent user interfaces can include such models as user models, discourse models and domain models [Maybury98] in order to drive adaptation. User models can be used to "tailor information presentation to the user, to predict the user's future behavior, to help the user find relevant information, and to adapt interface features to the user" [Maybury98, page 325].

2.3.1 User Models

Wahlster and Kobsa believe that the user model stemmed from the special purpose natural language interfaces of the 70s and the need for these systems to exhibit cooperative dialog behavior. "A cooperative system must certainly take into account the user's goals and plans, his/her prior knowledge about a domain, as well as false conceptions a user may possibly have concerning the domain" [Wahlster89, page 4].

There are many varying definitions of what a user model is. One definition by Belge and Ehrlich is "a set of concepts and metaphors devised by the designer to help the user understand the system" [Belge96, page 421]. They believe that the model can be created unconsciously by a formal method of the designer's choosing [Belge96]. Crow defines a user model as "any information which a program has which is specific to a particular user. The information itself could range from a simple count of errors, to some complicated data structure which purports to represent a relevant part of the user's knowledge of the problem domain" [Crow93, page 99]. Kass and Finin believe that user models are only beneficial to a system that seeks to adapt its behavior to individual users, or assumes responsibility for or with the user, or has a diverse potential set of users [Kass86].

Elaine Rich's article on the subject of user modeling is perceived as marking the beginning of research in the field [Maybury98]. She states that "it has long been recognized that in order to build a good system in which a person and a machine cooperate to perform a task it is important to take into account some significant characteristics of people. The system can then be designed to take advantage of those characteristics, rather than fight against them" [Rich83, page 199]. She believes that user models are necessary because they affect several factors that contribute to the ease of use of computer systems, such as the speed and quality of response as well as the language interface [Rich79].

Rich believes that stereotypes are useful in building such systems and describes them as "clusters of characteristics" [Rich79, page 330]. They are similar to the ideas of scripts, frames and schemas. There are two types of information a system must know in order to use a stereotype effectively, the stereotype itself and its facets such as the level of user expertise, as well as the triggers that signal the appropriateness or use of a particular stereotype. She states that computers have no emotional attachment to their stereotypes so there is no danger as in the use of human stereotypes. [Rich79]

Crow and Smith, on the other hand, disagree with Rich and believe "for any particular stereotype there is no such thing as a stereotypical user" [Crow93, page 98] resulting in inappropriate stereotypes. They believe that evidence proves that users vary too much for stereotypes to be successful and, therefore, a system that uses stereotypes "will at best be inadequate and at worst produce systems so ill-matched to their actual users that they will impede rather than assist them in getting their work done" [Crow93, page 98]. Their solution is an individual approach that builds a model of the user's tasks and looks for patterns for each individual user. They implement this solution in their adaptive interface system DB_Habits. [Crow93]

Rich however, uses stereotypes, information from the user and inferences from the user's actions to build the User Synopsis to guide the system. She associates with each piece of information a rating representing how confident the system is in the inferred knowledge. [Rich79]

Rich defines the difference between explicit and implicit models, explicit models allowing the user to create their own models explicitly, and implicit models taking charge of the personalization on its own. She believes that "people are not reliable sources of information about themselves" [Rich83, page 202], and therefore advocates the implicit approach. "People do not want to stop and answer a large number of questions before they get on with whatever they are trying to use the system to do. This is particularly true of people who intend to use the system only a few times, and for only brief periods" [Rich83, page 203]. In order to build a useful model, she proposes constructing a dictionary of system commands and associating with each the information its use provides about the user. [Rich83]

"Modeling the user's expertise is particularly important in help systems" [Oppermann94, page 85]. He sites Chin's work in which a user is classified as a novice, beginner, intermediate or expert in their knowledge of UNIX commands. He believes this level of modeling may be sufficient when the overall level of expertise is all the adaptive component needs. [Oppermann94]

A general user model, called GUMS, was devised by Kass and Finin with the purpose of designing multiple systems or being used by a wide variety of applications. The three types of user modeling facilities GUMS provides are the representation and maintenance facilities, access facilities and acquisition facilities. The representation facilities work with the user's goals and plans, the access facilities provide information about the users themselves and the acquisition facilities are used for acquiring knowledge about the user. [Kass91]

Similar to Rich's belief of implicit acquisition, GUMS uses a cooperative advisory system that is helpful and advises the user. There are four methods used by GUMS to acquire information about user's beliefs: the user's observable behavior, the system's behavior, the system's domain model and the current user model. Throughout their research of GUMS, the researchers found the idea of general user modeling to be feasible as well as practical. [Kass91]

2.4 Proficiency of the User

Benyon and Murray state that when "users change behavior as their experience with a system develops it may be expected that there will be a need for different interfaces for the same user and task at different stages" [Benyon93, page 115].

Bonar and Liffick take this expectation into account when they ask the very important question of how to "build a powerful and productive interface that will satisfy both experienced and novice computer users" [Bonar91, page 130]. "A interface that merely matches the user's expectations is stuck with those expectations. In particular, the user can never go beyond those expectations to use more powerful facilities than that expectation allows" [Bonar91, page 132]. They believe it is important to "build interfaces that allow graceful progression from the novice's use of a system to more sophisticated use of a system" [Bonar91, page 132].

In order to satisfy both of these types of users they use an approach that focuses on building "a series of usable system elements that, while complete at a certain level of functionality, also provided a scaffolding for higher levels of functionality" [Bonar91, page 132]. Their implementation of Bridge, a programming environment that teaches Pascal, allows the user who may be more sophisticated through experience with the program to recreate his own plan set. These plans are organized around the interests, intentions and experience of many different types of users. The plans are knowledge structures that capture the experience and intentions of the user or domain expert. [Bonar91]

Paris implements user knowledge in her system TAILOR that generates descriptions of devices such as telephones and disk drives. "Depending on the user's assumed domain knowledge, a description can be either parts-oriented or process-oriented. Thus the user's level of expertise in a domain can guide a system in choosing the appropriate facts from the knowledge base to include in an answer" [Paris89, page 200]. The researchers choose two distinct descriptive strategies, taking descriptions from adult encyclopedias, which are more parts oriented, and junior level encyclopedias, which are more process oriented, and merged the descriptions to accommodate users who fell between the levels of expert and novice. [Paris89]

The COACH system is a system that records user experience in order to create more personalized help files. It creates an adaptive user model from observing the user's actions and constructs help files on the basis of user-demonstrated experience and proficiency. COACH uses an advisory-style agent whose goal is to educate the user. Selker describes his advisory style agent in terms of the parable, "give a person a fish and you've fed them once, teach a person to fish and you have fed them for life" [Selker94, page 93]. He makes the association of the assistant-style agent and the fish, and the advisory style agent and teaching the user to fish.

Selker rates experience by keeping track of how many times a learnable thing has been used. In order to monitor the user's expertise Selker defines four levels with different characteristics. They are novice, intermediate, professional and expert. In the novice level examples are very simple and basic, in the intermediate level information is provided to help users know how and when they can use the learnable thing, in the professional level the information shows the available uses of the learnable thing, and in the expert level descriptions are like those seen in a reference manual. [Selker94]

Zellermayer et al. devised a system called the Writing Partner that helped students write papers by cueing them with unsolicited advice or solicited advice. They found that the students who were cued with unsolicited advice took longer to write their essays and did not show initial improvement, however, two weeks later, the advised students wrote better essays than the others [Zellermayer91]. Selker's study on the other hand, showed evidence that "unsolicited help can shorten rather than prolong a task" [Selker94, page 95].

2.4.1 Gathering Data on Expertise

Meyer, Yakemovie and Harris believe that speed and accuracy data can both be used for measuring user expertise. When using speed data, the system could measure how quickly a user completes certain tasks. However, there may be several considerations that may affect the data, for instance, system speed, hardware conditions, or user actions. For these reasons, they believe accuracy data is more useful for these kinds of measurements "particularly since the intelligence of an adaptive interface may be more effective in preventing common errors than in speeding up correct performance" [Meyer93, page 253]. However, obtaining this type of data is more difficult than obtaining speed data and is quite challenging. [Meyer93]

Desmarais and Liu believe that the lack of knowledge assessment in commercial applications is due to the "unavailability of simple and efficient knowledge techniques that non-specialists of AI/cognitive-modeling can use while developing their applications" [Desmarais93, page 308]. They propose a technique that uses a set of KUs (knowledge units) to represent the knowledge domain. The knowledge of the domain is modeled by numerical values, or weights, being attached to the nodes representing the likelihood of the user knowing a specific knowledge unit. The KUs are related by precedence relations. Observations are made about a user's knowledge state by question and answer sessions and implications are made when the knowledge of one KU implies knowledge of another KU. [Desmarais93]

Kelly et al. profess that "few research articles devoted to descriptions of rating schemes for measuring user proficiency exist" [Kelly98, page 34]. Their definition of a proficiency measurement is "any measurement of ability to complete work in a timely fashion and with few errors" [Kelly98, page 35]. They cite two methods of collecting data. For benchmark tasks it would be the percentage of accurately completed tasks, time to complete each task, and errors made while working on the task. At the other end, a user could be observed while completing the task. The first measurement is the one they use in their studies. Their studies consisted of teaching basic word processing notes and taking performance measures. As a result, they found that time and accuracy measures worked extremely well in providing these measures. [Kelly98]

Beck, Stern and Woolf focus on the creation of a student model that collects information about a student's problem solving ability, the acquisition of new concepts, and the student's retention of the old. The program is a mathematics tutor and it works by providing the students with hints. In order to achieve providing relevant hints, the student model must update itself. One way it does this is to examine the hints that the student needed in order to solve the problem, the student's current ability, and their acquisition and retention levels. In examining the hints, it considers the highest level hint that the student needed in order to complete the task. [Beck97]

Murphy and McTear focus on the design of an application called CASTLE that "takes into account the strengths, weaknesses, preferences and level of proficiency of each individual student when tutoring" [Murphy97, page 301]. There are four stereotype groups, novice, beginner, intermediate and advanced, and five proficiency levels of 0, 1, 2, 3, and 4. In order to update the proficiency level, a score is given based on the correctness of the student's answer to questions and how many times the user accesses the help facility. In addition, the Implicit Acquisition Rules component works by inferring proficiency by relating topics to topics that have already been learned, thus updating the proficiency of that topic.

The overall proficiency is calculated from the student's proficiency in all the completed topics. When the student makes an error it is mapped to topics in the student model. The number of errors is recorded and if the student makes three errors a remedial exercise is recommended. In addition, the system keeps records on each student so when the student reenters the program they are able to see a summary of topics covered and choose a new topic to explore. The system will then decide if the student may explore the new topic based on their proficiency level. [Murphy97]

2.5 Tutorials and Help

Tutorials and help systems are useful in guiding the user's interactions. There are many approaches to the development of these aids.

2.5.1 Tutorials

Dryer discusses two intelligent user interface technologies that aid the user by assisting them through the completion of tasks: wizards and guides. The difference between the two are that wizards are best for guiding the user through tasks that are completed infrequently; guides on the other hand are useful for more frequent tasks and they can help the user learn how to complete the task. Guides work best at educating the user about the interface or the task. These agents are best applied when the user is trying to complete a difficult or important task. Dryer found that "experience level did not significantly influence people's perceptions of the tasks" [Dryer97, page 267], thus was not a consideration in his evaluations of these technologies. [Dryer97]

The purpose of the study of Barnett et al. was to describe the framework and design considerations of implementing a tutorial. They relied heavily on the use of slides defined as "a collection of information that is "visible" at a given time" [Barnett98, page 87]. The authors suggested a linear organization for the slides with the addition of links, similar to Hypertext links in HTML. The links would allow the user to link to additional relevant slides. Each slide has a unique ID associated with it to allow sequencing of the slides. This sequencing was implemented by using a state transition table containing information concerning the slide ID, the event and the destination slide. [Barnett98]

2.5.2 Help Systems

Dicks discusses two approaches to developing and presenting information to users. The two approaches are to develop the help, documentation and training separately, then make them appear to be integrated, and to develop it as one set of information but allow it to be accessed in pieces, such as using a table of contents, key words, indexes and hyperlinks. He proposes that it is possible to develop information only once and use it for the purposes of help, documentation and training.

The method for accomplishing this integration was to structure the information based on the user's tasks. Dicks states that "[much] of the literature tells us to do this, and it seems intuitively obvious" [Dicks94, page 116]. In order to reuse the information the tasks were broken down into very small units and organized into chunks that consisted of the cases, the tasks and the steps. Dicks also points out that "effective learning support should be visually oriented" [Dicks94, page 116]. Another interesting point that he makes is that some people only feel secure when they have a hard copy of the documentation, thus a print function should always be provided to address the needs of these users. [Dicks94]

"Automatic help generation is widely recognized as an important feature in order to provide usable environments" [Pangoli95, page 181]. However, due to poor semantic support, help systems usually suffer and users are unable to associate information with the tasks they want to perform. Pangoli suggests obtaining automatic task oriented help from the user interface specification and structuring help by user tasks. This structure can be accomplished by task decomposition and the use of a task tree that can be navigated. Questions regarding the task, such as why the task is not allowed and if and how it can be performed, can be answered.

In order to structure the help message itself, Pangoli suggests using "pieces of prewritten text which are joined together by the help engine to form a sensible explanation" [Pangoli95, page 185]. When several answers are possible the 'or' connection should be used. [Pangoli95]

Knabe presents the origins of the Apple Guide, the online help system for Macintosh. The work by Apple's Human Interface Group was highly influential. In one study, using researchers to observe users thinking out loud while performing tasks with the HyperCard application, it was noted that the questions were divided into 5 distinct categories. These categories included goal, descriptive, procedural, interpretive and navigational questions. In another study, it was discovered that users preferred the design of the access screen to be similar to the contents of a book. The left side contained tasks and the right side contained sub-tasks related to the chosen task on the left.

In addition, Apple's instructional products group tested several different help access screen models. They included a topics screen containing broad topic categories, an index screen, allowing the user to click on an alphabetical list of topics, and a Look For screen that allowed users to search using keywords. The results of this study showed that novices preferred the topics screen, while more advanced users preferred the index and Look For screens.

As a result of these studies Apple's design goals for their help system were: help should appear in the same layer as the application itself, information should be presented in small chunks in order to enable the user to avoid having to rely too much on his memory, the system should send the user back to an instruction that was not completed, and when an instruction has been completed, the system should skip over it from that point on. [Knabe95]

2.6 Summary

The literature has indicated the importance of intelligent user interfaces that support the user as opposed to being a hindrance. An adaptive interface is preferable, as it changes dynamically while inferring the user's plans and goals. Several adaptive systems have been studied, such as CHORIS and the User Interface Design Environment.

User models, with their use of stereotypes, are an effective means of driving adaptation in an intelligent user interface. Implicit user models are preferred since people are not always the best sources of information about themselves and may not want to take time to provide this information [Rich83].

User proficiency can be used as a means for selecting an appropriate model, as it is important to provide different interfaces for different tasks at different levels of user proficiency [Benon93]. The COACH system is an example of such a system. Several methods exist for discerning user proficiency, such as speed and accuracy data and data related to the amount of times help is requested.

Chapter 3

ER-BY-DESIGN VERSION 2.0

User models, discussed in Chapter 2, are a widely accepted means of interface customization and much research has been completed in this area. The importance of taking into account the changing needs of users as their experience with a system's domain develops has also been researched quite extensively. However, research concerning a system that places the user in the appropriate user model dynamically, by monitoring the user's interaction with the system, is lacking. This chapter presents a research approach that involves a system that utilizes user models and dynamically monitors user interaction, in order to gauge user expertise, placing the user in the most appropriate model when it has been determined that user expertise has elevated, through use of the system and an increasing mastery of concepts from the application domain.

3.1 Overview

The adaptive version of ER-by-Design, referred to as ER-by-Design version 2.0, is designed to allow the user to create an Entity-Relationship diagram consisting of entities, relationships and attributes, while monitoring the user's interaction with the system and collecting data regarding this interaction to use in deciding if the user's proficiency in the creation of Entity-Relationship diagrams has increased, thereby placing the user in a more appropriate user model. This user model determines the current application interface based on the user's determined level of proficiency.

ER-by-Design version 2.0 consists of five parts: an application, ER-by-Design, created by Dr. Krissten N. Cooper, Lisa Hunt and Sue Petersen, that allows the user to create an Entity-Relationship diagram by leading them through a series of steps; an inference component that collects and analyzes data regarding the specific user's system usage, in order to place users in the most appropriate user model based on their perceived expertise in the application domain; a help system consisting of terminology and examples pertaining to ER modeling concepts; a help/assistance screen that presents task related information to the user; and the four user models responsible for customizing the interface for the user.

3.2 ER-by-Design

ER-by-Design is a learning tool designed to lead the user through the creation of a visual Entity-Relationship (ER) diagram. The initial version of ER-by-Design was intended primarily for the introductory level student.

The Entity-Relationship model was conceived by Peter Chin and is a popular, high level conceptual data model frequently used in the conceptual design of database applications. The ER model incorporates the concepts of entities, relationships and attributes. An entity can be an object with a physical or conceptual existence, relationships are objects that define an association between various entities. Attributes are specific properties of entities or relationships [Elmasri94].

There are several different types of attributes, such as a composite attribute that can be divided into smaller subparts, a multi-valued attribute that can have many different values and a derived attribute whose value can be derived from another attribute, or a related entity. Most entities have a unique attribute whose values are distinct; these types of attributes are known as key attributes. Entities that do not have a key attribute are known as weak entities and can only be identified by their relationship to another entity. If members of the same entity type participate more than once in a specific relationship then that relationship is known as a recursive relationship. The two types of constraints on relationships are cardinality, which specifies the number of times an entity instance can participate in a particular relationship, and participation, which specifies whether the existence of an entity depends on its having an instance of a specific relationship to another entity. These concepts are utilized in ER-by-Design. [Elmasri94]

ER-by-Design consists of a graphical interface that allows the user to select options by choosing appropriate menu choices (buttons) on the interface. The options are to create an entity, attribute or relationship, delete an entity, attribute or relationship, display an entity with its attributes, or a relationship with its attributes, and to draw a visual diagram that can be printed of the ER diagram with entities, relationships, cardinality and participation defined.

If the user chooses to create an object, such as an entity, relationship or attribute, the application leads them through the steps of the creation of the object by presenting the user with dialog boxes prompting the user to enter information. If the user chooses to view an entity or relationship and its attributes the application presents the user with a list of existing entities or relationships and once the appropriate one is chosen, displays the name of the object with the names of its attributes listed below. In order to delete an object the user is asked to choose the type of object they would like to delete (entity, relationship or attribute), then given a list of the existing objects of the chosen type and allowed to select which object to delete. If the user chooses to delete an entity, all attributes and relationships of that entity will be deleted as well, if the user chooses to delete a relationship, all attributes of that relationship will be deleted and if the user chooses to delete a composite attribute, consisting of sub-attributes, the sub-attributes are deleted as well.

The application also enables the user to draw the finished diagram with a draw screen module that allows the user to drop the entity or relationship on the screen and also display the cardinality and participation for each relationship and its corresponding entities as specified by the user. The application allows the user to save the ER design sessions to a data file, enabling them to return to and continue working on previous ER designs.

3.3 Enhancements to ER-by-Design

The enhancements of the initial version of ER-by-Design, which is the focus of this thesis, are the inference component, the help system, the help/assistance screen and the user models. These enhancements were created in order to create a system that allows adaptation of the interface whenever the user has demonstrated a higher level of proficiency in the application domain of ER modeling. The application enhancements were designed in Visual Basic 5.0 and currently operate on an IBM-compatible personal computer under the Windows 95/98 operating system.

3.3.1 Inference Component

The inference component is responsible for monitoring the interaction between the user and the system, collecting statistical data regarding this interaction, and determining if the user's expertise in the application domain of ER modeling has elevated significantly. The metrics implemented by the inference component that determines whether adaptation should take place by representing the user's proficiency with the system, are the speed in the completion of set tasks, the user's use of the help system, the number of times the user chooses the rollback option and the complexity of the completed ER diagram. If the results of each of these measurements are significant, then the inference component will determine that the user's expertise has elevated significantly to merit promotion to a more advanced user model.

3.3.1.1 Speed

Much research has supported the use of speed in the completion of tasks as a measurement of user proficiency. [Meyer93] [Kelly98] In determining the average speed of the completion of a set task, a set task being the creation of an entity, relationship or attribute, each task is weighted according to the length of time it would take the average user to complete it. The reason for the applied weights is due to the fact that it takes users on average longer to complete the task of creating certain objects as opposed to others. For this reason a weight of five is applied to the time it takes to create an entity, a weight of two is applied to the time it takes to create an attribute and since on average, the task of creating a relationship takes longer to complete, a weight of one is applied to the time it takes to create a relationship. The averages of the three sub-tasks are then used to determine the average speed of the completion of an overall task.

The system uses the metric of the average speed of the completion of an overall task because the user may not complete all of the three sub-tasks in one session and may complete a different set of tasks in different sessions. By weighting the sub-task speeds and averaging them into an overall task speed, a uniform speed measurement is collected for each session.

Since overall task speeds can vary to extremes due to environmental factors out of the researcher's control, averages are taken of the overall task speeds for the first two sessions and the last two sessions for a specific user model. The result is that the user must have completed at least three sessions in a particular user model in order for the inference component to evaluate if the user's level of expertise has progressed across sessions. The average speed of the last two sessions must be greater than a twenty-five percent increase over the average speed of the first two sessions to be significant.

3.3.1.2 Help

The number of times a user accesses the help system and the level of help accessed is also a useful measurement of user expertise and has been used by Beck, Stern and Woolf, and Murphy and McTear in terms of their respective applications the mathematics tutor and CASTLE [Beck97] [Murphy97]. Although Beck, Stern and Woolf use a hint system, monitoring how many times the user needs hints and the level of hints requested, the system is similar to ER-by-Design version 2.0's help system in that the hints are weighted and these weights are taken into account when the help measurement is analyzed by the system.

The help system is divided according to the following categories, beginner help, intermediate help and advanced help. Due to her experience in teaching database concepts, Dr. Krissten N. Cooper, was consulted in placing help concepts, contained in the help system, into categories. Help concepts classified as beginner concepts are given a weight of sixteen, help concepts classified as intermediate concepts are given a weight of twenty and help concepts classified as advanced concepts are given a weight of twenty-five. Each weight is a twenty-five percent increase from the weight of the help concept that precedes it.

In order to calculate a help score, representing the user's level of help usage, the help weights are added together and divided by the number of times the user accesses help in a particular session. The measurement of the level of help accessed is most valid and informative for users who use help in a consistent manner. For this reason, help usage is only counted as a measurement for a session in which the user accesses help at least three times. If the user does not access help to this degree, the help score for the current session is given a value of zero. Similarly to the speed measurement, averages of the use of help are created by averaging the help score from the first two sessions and averaging the help score from the last two sessions. This average results in a more uniform and realistic help score to use as a measurement of user expertise. In order to be significant, the average help score of the last two sessions must be more than a twenty-five percent increase over the average help score of the first two sessions, representing a significant increase in the level of help accessed.

The help measurement is excluded as a measurement of user expertise under certain conditions. These conditions include the user not accessing the help system for either of the first two sessions, or the result of a help score, reflecting an average of the first two sessions, that is greater than twenty. The measurement is intended to measure the increase in the level of help of a user who uses help in a consistent manner, so users who do not use help in the first two sessions should not have their expertise measured by this metric. If the user's help usage results in a help score of twenty or greater, as an average for the first two sessions, then the user is using advanced help and can not increase their level of help from this help score significantly.

3.3.1.3 Rollbacks

The rollback option can be used while the user is completing one of the three sub-tasks, creating an entity, relationship or attribute. This feature allows the user to "rollback" to the option of beginning the same sub-task again or creating a new sub-task, without saving the information that was entered for the current sub-task. It is a useful option when the user becomes confused or realizes that they are not creating the sub-task correctly. For this reason, the number of times the option is used is a useful measurement of user proficiency in the domain of ER design, as it will provide data pertaining to the level of difficulty or confusion with completion of sub-tasks.

The rollback score is used to reflect perceived user proficiency. The rollback score is calculated by dividing the number of times that the user chose the rollback option by the number of completed sub-tasks. Completed sub-tasks are sub-tasks that have been completed without using the rollback option. Unlike the speed and help metric, the rollback score is not averaged for the first two sessions and last two sessions in a current model and is not examined in relationship to the previous rollback score. The rollback score is examined for its significance in the current session only.

The rollback score is significant for different users based on their level of expertise represented by their current user model. The rollback score is significant for the user placed in the beginner user model if it is less than one, implying that a beginning user may be able to be promoted to a more advanced user model if the number of times they access the rollback option is less than the number of times they complete a sub-task. A rollback score is significant for the user placed in the intermediate user model if it is less than .5, implying that an intermediate user may be able to be promoted to a more advanced user model if the number of times they access the rollback option is less than fifty percent of the number of times they complete a sub-task.

3.3.1.4 Complexity

As a user's proficiency in ER modeling concepts increases, it should be reflected in the complexity of the ER models being created by the system. For this reason, a complexity measurement is calculated and analyzed by the system as a measurement of user expertise. The complexity score formula is displayed in Figure 1.

Complexity Score =
Average Relatedness Count *
Number of Entities and Relationships

Average Relatedness Count =
sum of relationships each entity participates in /
sum of entities

Number of Entities and Relationships =
sum of entities + sum of relationships +
sum of weak entities + sum of recursive relationships

Figure 1: Complexity Score Formula

In the complexity score formula, since weak entities and recursive relationships imply a more advanced level of knowledge they are basically counted twice, for instance a weak entity is counted once as an entity and again as a weak entity.

Similarly to the rollback measurement, the complexity score is examined for its significance in the current session only and the complexity score is significant for different users based on their level of expertise represented by the current user model. The levels of significance were determined by consultation with Dr. Krissten N. Cooper and by examination of ER diagrams appropriate for users at different levels of proficiency. For the user placed in the beginner user model, a complexity score greater than fifteen is significant, for the user placed in the intermediate user model, the complexity score is significant if it is greater than twenty.

3.3.2 Help System

The help system was created to aid the user in learning concepts relevant to ER design as well as providing system specific information. The help system consists of several help modules as well as help topics available from the menu.

The tutorial module encompasses an overview of the entire process of ER design by presenting an example ER problem and its solution, educating the user on the concepts of entities, relationships and attributes, how they are related, and leading the user through the steps of creating an ER design. The tutorial module also presents the user with the completed visual ER diagram. The terminology module allows the user to select a concept from the list and then displays its definition. The symbol lookup module is provided to familiarize the user with the different symbols used in the process. This module was included since an important factor in learning to create ER diagrams is knowledge of the symbols used. Screen captures of the tutorial, terminology and symbol lookup module are included in Appendix A.

In addition to these modules, the help system consists of additional topics available to the user from the menu. The help menu is designed in tree structure form so the user may begin with a general concept and narrow the search down to more refined concepts, with several clicks of the mouse.

3.3.3 Help/Assistance Screen

Dicks and Pangoli both discuss the importance of providing information based on the user's tasks. In addition Pangoli believes that help systems suffer because they are unable to associate information with the tasks to be performed.

[Dicks94] [Pangoli95]. This concept of providing information based on the user's task was considered in the design of the help/assistance screen. The purpose of the help/assistance screen is to enhance learning or reinforcement of the concepts used in ER design and to aid the user in completion of the chosen task.

The help/assistance screen is displayed whenever the user is in the process of creating an entity, relationship or attribute. The screen is a large slide similar to Barnett's slide concept defined as a "collection of information that is visible at a given time" [Barnett98, page 87]. The help/assistance screen is displayed in the right one half of the window containing the application. The dialog boxes that lead the user through the steps of completing the tasks appear on the left side of the window. The help/assistance screen was designed to be large enough to draw the attention of the user and there are no available options to hide the screen from view.

The help/assistance screens are primarily composed of verbiage that explains information pertaining to the current step in the task at hand by presenting the user with definitions of concepts as well as examples. Graphics are also provided to enhance the learning or reinforcement experience as well as links to the help system. These links are available to take the user to similar help topics related to the current concept the user is working with.

The help/assistance screens are coordinated with the dialog boxes that lead the user through the task. Each time a new step is presented to the user, by way of a new dialog box, the help/assistance screen will alter its contents to provide information pertaining to the current step of the current task. A screen capture that includes the help/assistance screen is included in Appendix B.

3.3.4 User Models

The user models are responsible for altering and aligning the system interface to correspond to the level of proficiency of the user. Several user models are available corresponding to several varying levels of user expertise.

The interface presented to the user consists of buttons designed to allow the user to select tasks to complete, dialog boxes used to query information from the user in steps, and a help/assistance screen used to provide the user with information regarding the task to be completed. The various user models differ in the design of the help/assistance screen presented to the user; however, because the steps in designing ER models are the same for all levels of users, the dialog boxes used to query information from the user remain the same in all models.

3.3.4.1 Naive

This model represents a user who is new to ER modeling concepts and/or are using this application for only the first or second time. If it is their first time to use the system, this category of user will be asked to create a login password. The user will be presented with the tutorial module, the terminology module and the symbol look up module. When the user has exited the three help modules, the application will begin leading the user through the process of building an ER diagram. Whenever a new task appears or a new concept arises, the help/assistance area will display information tailored to the current task as well as to the user model. For the naïve user the help/assistance area will contain an explanation of the current task, specific and basic definitions of concepts relating to the current task, as well as links to specific examples and help files relating to the current task. When the user makes a choice or finishes a specific task, an explanation will be displayed detailing exactly what that choice represents in terms of the ER design. This information will be used to reinforce the concepts in the user's mind. The choice of wording and the detail in this area will be quite basic and in detail.

3.3.4.2 Beginner

This model represents a user who is at the same level of expertise as the user placed in the naive model, with the exception that they have used the application two or more times and, therefore, should be more familiar with how the program works. Because of this familiarity, this user will no longer automatically be shown the tutorial, terminology and symbol look up modules. However, as in all models they are free to examine this material, indexed in the help section at any time. Other than these differences, the model will be the same as the naïve user model. Screen captures including the creation of an entity, the creation of a relationship and the creation of an attribute in the beginner user module are included in Appendix C.

3.3.4.3 Intermediate

This model represents a user who is quite familiar with the concepts of the ER model but would still benefit from links to examples and less detailed and intense definitions. This model would also be suitable for a user who is knowledgeable about ER concepts but has not applied them recently. This user will not be required to view any of the help modules and will be shown a help/assistance area containing definitions of concepts relating to the current task with less detailed dialogs than the above models. Also displayed will be links to specific examples and help files relating to the current task. When a user makes a choice or finishes a certain task, a brief explanation will be displayed confirming the choice that has been made. Screen captures of creating an entity, creating a relationship and creating an attribute in the intermediate user module are included in Appendix C.

3.3.4.4 Advanced

This model represents a user who is extremely proficient with the concepts and design of an ER model, and who does not need lengthy descriptions in the help/assistance section and would be distracted by them. The only displayed information in the help/assistance section is brief descriptions, links to specific examples and help files relating to the current task. Screen captures of creating an entity, creating a relationship and creating an attribute in the advanced user module are included in Appendix C.

3.4 Summary

The first time an individual uses ER-by-Design version 2.0, they will be placed in the naïve user model. After successfully creating two ER models with the application, the user will be moved to the beginner model and will subsequently be moved to more advanced models based on system interaction. Each user is required to stay in the current model, with the exception of the naive model, at least three times before promotion to a more advanced model, due to the metrics involved in assessing user proficiency. These metrics are the speed in the completion of set tasks, the user's use of the help system, the number of times the user chooses the rollback option and the complexity of the completed ER diagram.

At the end of each session, the inference component will statistically analyze the metrics for promotion, in order to assess if the user's proficiency has elevated significantly. If enough sessions have been completed and the results of the statistical analysis conclude that user proficiency has elevated to a significantly higher level, the user will be promoted in the next session. Upon entering the application for the next session, the user will be notified that they have been promoted to a new user model and will continue using the program with a new interface based on the new user model. The interfaces for each user model consist of varying help/assistance screens that display information based on the user’s current task.

Chapter 4 analyzes the performance of ER-by-Design 2.0, examining measurements concerning the efficiency and effectiveness of the system. In order to accurately ascertain the efficiency and effectiveness of ER-by-Design version 2.0, measurements relating to relative improvement in task completion time, relative improvement in amount of help usage, the complexity and correctness of the final ER diagrams, and user perception of the system over time, are collected and statistically analyzed.

Chapter 4

ANALYSIS OF ER-BY-DESIGN VERSION 2.0

In order to analyze the performance of ER-by-Design version 2.0, three experiments are conducted comparing ER-by-Design version 2.0 to a static system that does not implement user models. To accurately determine the efficiency and effectiveness of ER-by-Design version 2.0 measurements relating to relative improvement in task completion time, relative improvement in amount of help usage, the complexity and correctness of the final ER diagrams, and user perception of the system over time, are collected and statistically analyzed.

4.1 Overview

To determine the efficiency and effectiveness of ER-by-Design version 2.0, the system was compared to a static version that did not incorporate user models and therefore, did not adapt to the user's proficiency level throughout use of the system. The static version had the same interface presented by the advanced user model of ER-by-Design version 2.0. This interface consisted of a help/assistance screen that provided only minimal information to the user regarding the current task as well as links to further information located in the help system. Consequently, the help/assistance screen provided by the static version of the system provided very little task-related help or assistance. Appendix D includes screen captures of ER-by-Design version 2.0 and the static version of the system.

4.2 Test Subjects

Undergraduate and graduate computer and information sciences students were selected to participate in the experiments. The undergraduate students were currently enrolled in a database course where basic ER design concepts were being taught. On the other hand, the graduate students were expected to have experience with ER design concepts whether at work or in the classroom.

The test group was to interact with ER-by-Design version 2.0 throughout the experiments and was expected to progress through several user models. This group consisted of eight graduate students and eleven undergraduate students, for a total of nineteen participants. The control group was to interact with the static version of the system and did not progress through any user models, always interacting with the same interface. This group consisted of seven graduate students and eleven undergraduate students, for a total of eighteen participants.

4.3 Experiments

Three experiments were conducted. Each experiment involved all participants. During an experiment, participants were given the same ER design problem and asked to build the ER design by using their assigned version of the system. Since there were three experiments, the participants were able to create at least three different ER designs with the systems. Since several metrics used to determine user proficiency were analyzed only after the third session in the same user model, as stated in the previous chapter, participants were asked to save and exit the program several times during each experiment to ensure that each participant had several sessions. A session was only determined complete if the user had successfully created at least five objects (entities, relationships or attributes) of at least two of the three major types.

The first experiment presented the participants with several simple, small ER design schemas to build with the version of the system they were assigned. The participants were asked to finish the exercises outside of the time allotted for the first experiment if they had not already done so. The primary purpose of the first experiment was to give the participants exposure to the systems. The ER diagram exercise for the first experiment is included in Appendix E.

The second experiment presented the participants with one ER design schema to build with their version of the system. The ER design exercise was at a moderate level of difficulty. The participants were timed during this experiment and had one hour to complete the exercise. The ER diagram exercise for the second experiment is included in Appendix E.

The third experiment also presented the participants with one ER design schema to build with their version of the system; however, this ER exercise was at a more advanced level of difficulty than the one presented in experiment two. Similarly to experiment two, the participants had one hour to complete the exercise. The ER diagram exercise for the third experiment is included in Appendix E.

4.4 Analysis of Results

The null hypothesis for comparing both versions of the system, in terms of the user's efficiency and effectiveness in interacting with the system, states that a dynamic system that adapts itself to a user's current state of expertise in terms of the application domain, by constantly monitoring the user throughout use of the system and placing them in appropriate user models when this expertise has changed, does not provide a more effective and efficient environment over time for users, when compared to a system with a static, generic interface based on a static user model. The alternative hypothesis states that a dynamic system that adapts itself to a user's current state of expertise in terms of the application domain, by constantly monitoring the user throughout use of the system and placing them in appropriate user models when this expertise has changed, does provide a more effective and efficient environment for users over time, when compared to a system with a static, generic interface based on a static user model.

In order to measure the user's efficiency and effectiveness in interacting with the system over time, metrics concerning relative improvement in task completion time, relative improvement in amount of help usage, the complexity and correctness of the final ER diagram and user perception of the system over time were monitored throughout the three experiments. At the end of each experiment, for each user, the system computed and recorded to a log file data concerning relative improvement in task completion time, relative improvement in the amount of help usage and complexity of the ER diagram. Correctness was computed manually by Dr. Krissten N. Cooper. Each user completed a user survey in order to ascertain user perception. The user surveys for the static version of the system and ER-by-Design are included in Appendix F.

Since participants were randomly assigned to one of the two groups and equal variance was not assumed, two-sample t-tests assuming unequal variance were used to compare the means of the two groups. The alpha level was set at .10

for all t-tests and results were acknowledged as statistically significant if the probability value was less than alpha.

4.4.1 Relative Improvement in Task Completion Time

The null hypothesis for comparing relative improvement in task completion time for the two versions of the system states that there is no difference in relative improvement in task completion time between the two groups. The alternative hypothesis states that the group interacting with ER-by-Design version 2.0 shows greater relative improvement in task completion time as compared to the group interacting with the static version of the system. The null and alternative hypotheses for comparing relative improvement in task completion time for the two versions of the system are represented in Figure 2.

$$H_o: \left(\left(\sum_{i=1}^{n(group\ 1)} (\Delta t_{i(group\ 1)} / \text{Experiment 1 } t_{i(group\ 1)})\right)/n\right) - \left(\left(\sum_{i=1}^{n(group\ 2)} (\Delta t_{i(group\ 2)} / \text{Experiment 1 } t_{i(group\ 2)})\right)/n\right) = 0$$

$$H_a: \left(\left(\sum_{i=1}^{n(group\ 1)} (\Delta t_{i(group\ 1)} / \text{Experiment 1 } t_{i(group\ 1)})\right)/n\right) - \left(\left(\sum_{i=1}^{n(group\ 2)} (\Delta t_{i(group\ 2)} / \text{Experiment 1 } t_{i(group\ 2)})\right)/n\right) > 0$$

group 1 = ER-by-Design version 2.0
group 2 = static system
Δt = Experiment 1 t - Experiment 3 t
t = task completion time

Figure 2: Null and Alternative Hypotheses for Relative Improvement in Task Completion Time

For each participant relative improvement in task completion time was calculated by the formula represented in Figure 3.

Relative Improvement in Task Completion Time = (Δt / Experiment 1 t)

Δt= Experiment 1 t - Experiment 3 t
t = task completion time

Figure 3: Relative Improvement in Task Completion Time

A two-sample t-test assuming unequal variance was performed to compare relative improvement in task completion time of the group interacting with ER-by-Design version 2.0 to relative improvement in task completion time of the group interacting with the static version of the system. The result of this test is displayed in Table 1.

ER-by-Design version 2.0	Static version	T-value	Reject (90% confidence)
MEAN: 0.33 STDEV: 0.38 n=19	MEAN: 0.24 STDEV: 0.90 n=18	-0.38	NO

Table 1: Result of T-test Performed on Relative Improvement in Task Completion Time

In order to determine if knowledge in ER design concepts was a factor that contributed to relative improvement in task completion time between users of the ER-by-Design version 2.0 and users of the static version of the system, the groups were divided into graduate participants, undergraduate participants, beginning level participants, intermediate level participants and advanced level participants in terms of user perceived proficiency. Self-rankings of the level of user proficiency were collected by the user survey after each experiment. The user defined levels of proficiency specified after the first experiment were used to divide the participants into beginning, intermediate or advanced level groups. The results of these tests are displayed in Table 2.

Group	ER-by-Design version 2.0	Static version	T-value	Reject (90% confidence)
Graduate participants	MEAN: 0.36 STDEV: 0.20 n=8	MEAN: 0.11 STDEV: 1.39 n=7	-0.89	NO
Under-graduate participants	MEAN: 0.31 STDEV: 0.49 n=11	MEAN: 0.47 STDEV: 0.29 n=11	0.93	NO
Beginning level participants	MEAN: 0.22 STDEV: 0.66 n=6	MEAN: 0.35 STDEV: 0.35 n=6	0.42	NO
Intermediate level participants	MEAN: 0.39 STDEV: 0.21 n=8	MEAN: 0.9 STDEV: 1.26 n=9	-0.70	NO
Advanced level participants	MEAN: 0.35 STDEV: 0.11 n=5	MEAN: 0.46 STDEV: 0.19 n=3	0.87	NO

Table 2: Results of T-tests Performed on Relative Improvement in Task Completion Time Based on Groups

The results of the t-tests indicate that the null hypothesis can not be rejected with greater than a 90% degree of confidence; therefore, it can not be concluded that the group interacting with ER-by-Design version 2.0

shows greater relative improvement in task completion time as compared to the group interacting with the static version of the system.

4.4.2 Relative Improvement in Amount of Help Usage

The null hypothesis for comparing relative improvement in amount of help usage for the two versions of the system states that over time there is no difference between the two groups in relative improvement in amount of help usage. The alternative hypothesis states that over time the group interacting with ER-by-Design version 2.0 shows greater relative improvement in amount of help usage as compared to the group interacting with the static version of the system. The null and alternative hypotheses for comparing relative improvement in amount of help usage for the two versions of the system are represented in Figure 4.

$$Ho: \left(\left(\sum_{i=1}^{n(group\ 1)} (\Delta h_{i(group\ 1)} / \text{Experiment } 1\ h_{i(group\ 1)})\right)/n\right) -$$

$$\left(\left(\sum_{i=1}^{n(group\ 2)} (\Delta h_{i(group\ 2)} / \text{Experiment } 1\ h_{i(group\ 2)})\right)/n\right) = 0$$

$$Ha: \left(\left(\sum_{i=1}^{n(group\ 1)} (\Delta h_{i(group\ 1)} / \text{Experiment } 1\ h_{i(group\ 1)})\right)/n\right) -$$

$$\left(\left(\sum_{i=1}^{n(group\ 2)} (\Delta h_{i(group\ 2)} / \text{Experiment } 1\ h_{i(group\ 2)})\right)/n\right) > 0$$

group 1 = ER-by-Design version 2.0
group 2 = static system
Δh = Experiment 1 h - Experiment 3 h
h = number of times help is requested

Figure 4: Null and Alternative Hypotheses
for Relative Improvement in Amount of Help Usage

For each participant relative improvement in amount of help usage was calculated by the formula represented in Figure 5.

Relative Improvement in Amount of Help Usage =
(Δh / Experiment 1 h)

Δh= Experiment 1 h - Experiment 3 h
h = number of times help is requested

Figure 5: Relative Improvement in Amount of Help Usage

A two-sample t-test, assuming unequal variance, was performed in order to compare relative improvement in amount of help usage of the group interacting with ER-by-Design version 2.0 to relative improvement in amount of help usage of the group interacting with the static version of the system. The result of this test is displayed in Table 3.

ER-by-Design version 2.0	Static version	T-value	Reject (90% confidence)
MEAN: 0.50 STDEV: 0.51 n=19	MEAN: 0.07 STDEV: 0.96 n=18	-1.70	YES

Table 3: Result of T-test Performed on Relative Improvement in Amount of Help Usage

The result of the t-test indicates that the null hypothesis can be rejected with greater than a 90% degree of confidence; therefore, it can be concluded that the group interacting with ER-by-Design version 2.0 shows greater relative improvement in amount of help usage as compared to the group interacting with the static version of the system.

Of interest were the effects of gender on the amount of help usage. It is theorized that females use help more often than males and, therefore, females may provide a more accurate measurement of relative improvement in amount of help usage. A two-sample t-test, assuming unequal variance, was performed in order to compare the average amount of help usage of females and males. The result of this test is displayed in Table 4.

Females	Males	T-value	Reject (90% confidence)
MEAN: 1.51 STDEV: 2.62	MEAN: 1.13 STDEV: 1.39	-0.43	NO

Table 4: Result of T-test Performed on Amount of Help Usage Based on Gender

The result of the t-test indicates that for this research, there is not a significant difference in the amount of help usage between females and males.

4.4.3 Complexity

The null hypothesis for comparing the complexities of ER diagrams created by the two versions of the system states that there is no difference in the complexity of the final ER diagrams created by the two groups. The alternative hypothesis states that the group interacting with ER-by-Design version 2.0 creates more complex final ER diagrams as compared to the group interacting with the static version of the system.

A two-sample t-test, assuming unequal variance, was performed in order to compare the average complexity of the final ER diagrams created by the group interacting with ER-by-Design version 2.0 to the average complexity of the final ER diagrams created by the group interacting with the static version of the system. The scores reflect a scale of 1 to 20 with 1 being the least complex and 20 being the most complex. The result of this test is displayed in Table 5.

ER-by-Design version 2.0	Static version	T-value	Reject (90% confidence)
MEAN: 15.32 STDEV: 3.85 n=19	MEAN: 14.37 STDEV: 4.18 n=18	-0.72	NO

Table 5: Result of T-test Performed on Complexity Scores of Final ER Diagram

Graduate, undergraduate, beginning level, intermediate level and advanced level groups were investigated. The results of these tests are displayed in Table 6.

Group	ER-by-Design version 2.0	Static version	T-value	Reject (90% confidence)
Graduate participants	MEAN: 15.56 STDEV: 3.40 n=8	MEAN: 12.57 STDEV: 6.27 n=7	-1.07	NO
Undergraduate participants	MEAN: 15.15 STDEV: 4.30 n=11	MEAN: 15.52 STDEV: 1.67 n=11	.26	NO
Beginning level participants	MEAN: 14.7 STDEV: 4.95 n=6	MEAN: 15.08 STDEV: 2.24 n=6	0.15	NO
Intermediate level participants	MEAN: 16.15 STDEV: 3.82 n=8	MEAN: 14.26 STDEV: 5.61 n=9	-0.81	NO
Advanced level participants	MEAN: 15.6 STDEV: 2.98 n=5	MEAN: 13.63 STDEV: 2.82 n=3	-0.93	NO

Table 6: Results of T-tests Performed on Complexity Scores of Final ER Diagram Based on Groups

The results of the t-tests indicate that the null hypothesis can not be rejected with greater than a 90% degree of confidence; therefore, it can not be concluded that the group interacting with ER-by-Design version 2.0 created more complex final diagrams as compared to the group interacting with the static version of the system.

4.4.4 Correctness

The null hypothesis for comparing the correctness of ER diagrams created by the two versions of the system states that there is no difference in the correctness of the final ER diagrams created by the two groups. The alternative hypothesis states that the group interacting with ER-by-Design version 2.0 creates more correct final ER diagrams as compared to the group interacting with the static version of the system.

After applying the steps of the algorithm for ER-to-Relational mapping, correctness scores were ascertained, on a scale of 0 to 100, by Dr. Krissten N. Cooper. ER-to-Relational mapping derives a relational database schema from a conceptual schema developed using the ER model [Elmasri94].

A two-sample t-test, assuming unequal variance, was performed in order to compare the average correctness scores of the final ER diagrams created by the group interacting with ER-by-Design version 2.0 to the average correctness of the final ER diagrams created by the group interacting with the static version of the system. The result of this test is displayed in Table 7.

ER-by-Design version 2.0	Static version	T-value	Reject (90% confidence)
MEAN: 76.57 STDEV: 12.25 n=19	MEAN: 73.61 STDEV: 14.12 n=18	-0.68	NO

Table 7: Result of T-test Performed on Correctness Scores of Final ER Diagram

Graduate, undergraduate, beginning level, intermediate level and advanced level groups were investigated and the results of these tests are displayed in Table 8.

Group	ER-by-Design version 2.0	Static version	T-value	Reject (90% confidence)
Graduate participants	MEAN: 74.37 STDEV: 10.83 n=19	MEAN: 72.14 STDEV: 17.28 n=18	-0.29	NO
Undergraduate participants	MEAN: 78.18 STDEV: 13.46	MEAN: 74.54 STDEV: 12.54	-0.65	NO
Beginning level participants	MEAN: 85 STDEV: 8.36	MEAN: 72.5 STDEV: 14.74	-1.80	YES
Intermediate level participants	MEAN: 71.25 STDEV: 12.46	MEAN: 72.77 STDEV: 15.83	0.22	NO
Advanced level participants	MEAN: 75 STDEV: 12.24	MEAN: 78.33 STDEV: 10.40	0.40	NO

Table 8: Results of T-test Performed on Correctness Scores of Final ER Diagram Based on Groups

The results of the t-tests indicate that the null hypothesis can be rejected with greater than a 90% degree of confidence for the beginning level group only; therefore, it can be concluded that the beginning level group interacting with ER-by-Design version 2.0 created more correct final diagrams as compared to the beginning level group interacting with the static version of the system.

4.4.5 User Perception

The null hypothesis for comparing user perception over time of the two versions of the system states that for the final experiment there is no difference in user perception between the two groups. The alternative hypothesis states that for the final experiment the group interacting with ER-by-Design version 2.0 shows more positive user perception as compared to the group interacting with the static version of the system.

User perception of the system was determined by the results of the user surveys that were completed after each experiment. The user survey included Questions 1 through 7. Scores for these questions reflected a scale of 1 to 7 with 1 being the least positive and 7 being the most positive level of perceived perception. The survey also included Question 8, that allowed the participants to rate themselves in terms of their user expertise and had an additional section for comments. Copies of the user surveys for the static version of the system and ER-by-Design version 2.0 are included in Appendix F.

A two-sample t-test, assuming unequal variance, was performed in order to compare user perception of the group interacting with ER-by-Design version 2.0 to user perception of the group interacting with the static version of the system for the final experiment. The result of this test is displayed in Table 9.

ER-by-Design version 2.0	Static version	T-value	Reject (90% confidence)
MEAN: 5.59 STDEV: 1.11 n=19	MEAN: 6.02 STDEV: 0.68 n=18	-1.39	YES

Table 9: Result of T-test Performed on User Perception for the Final Experiment

The results of the t-test indicates that the null hypothesis can be rejected with greater than a 90% degree of confidence; therefore, it can be concluded that for the final experiment the group interacting with ER-by-Design version 2.0 shows more positive user perception as compared to the group interacting with the static version of the system.

Two-sample t-tests, assuming unequal variance, were also performed in order to compare user perception of the group interacting with ER-by-Design version 2.0 to user perception of the group interacting with the static version of the system for each question for the final experiment. The results of these tests are displayed in Table 10.

Question	ER-by-Design version 2.0	Static version	T-value	Reject (90% confidence)
1	MEAN: 6.05 STDEV: 0.91 n=19	MEAN: 5.61 STDEV: 1.24 n=18	-1.22	NO
2	MEAN: 6.15 STDEV: 0.68 n=19	MEAN: 5.88 STDEV: 1.13 n=18	-0.86	NO
3	MEAN: 5.68 STDEV: 1.05 n=19	MEAN: 5.27 STDEV: 1.36 n=18	-1.00	NO
4	MEAN: 6.10 STDEV: 0.73 n=19	MEAN: 5.61 STDEV: 1.37 n=18	-1.34	YES
5	MEAN: 6.10 STDEV: 0.80 n=19	MEAN: 5.66 STDEV: 1.13 n=18	-1.34	YES
6	MEAN: 6.05 STDEV: 0.91 n=19	MEAN: 5.44 STDEV: 1.38 n=18	-1.57	YES
7	MEAN: 6.00 STDEV: 0.81 n=19	MEAN: 5.60 STDEV: 1.13 n=18	-1.01	NO

Table 10: Results of T-tests Performed on Improvement in User Perception Based on Each Question for the Final Experiment

The results of the t-tests indicate that the null hypothesis can only be rejected with greater than a 90% degree of confidence for Questions 4, 5 and 6; therefore, it can be concluded that for the final experiment the group interacting with ER-by-Design version 2.0 shows more positive user perception as compared to the group interacting with the static version of the system for Questions 4, 5 and 6.

Investigating the amount of improvement in user perception can also result in further information. For each participant improvement in user perceived proficiency was calculated by the formula represented in Figure 6.

Improvement in User Perception = Δp

Δp = Experiment 3 p - Experiment 1 p
p = user perception

Figure 6: Improvement in User Perception

A two-sample t-test, assuming unequal variance, was performed in order to compare improvement in user perception of the group interacting with ER-by-Design version 2.0 to improvement in user perception of the group interacting the static version of the system. The result of this test is displayed in Table 11.

ER-by-Design version 2.0	Static version	T-value	Reject (90% confidence)
MEAN: 0.21 STDEV: 0.71 n=19	MEAN: -0.52 STDEV: 0.89 n=18	-2.89	YES

Table 11: Result of T-test Performed on Improvement in User Perception

Analyzing improvement in user perceived proficiency was also an important factor in determining user perception of the system over time. Question 8 allowed each user to rate their user expertise in designing ER diagrams. Scores for Question 8 reflected a scale of 1 through 7 with 1 being a rating of beginner, 4 being a rating of intermediate and 7 being a rating of advanced. For each participant improvement in user perceived proficiency was calculated by the formula represented in Figure 7.

Improvement in User Perceived Proficiency = Δr

Δr= Experiment 3 r - Experiment 1 r
r = user perceived proficiency in creating ER diagrams

Figure 7: Improvement in User Perceived Proficiency

A two-sample t-test, assuming unequal variance, was performed in order to compare improvement in user perceived proficiency of the group interacting with ER-by-

Design version 2.0 to improvement in user perceived proficiency of the group interacting the static version of the system. The result of this test is displayed in Table 12.

ER-by-Design version 2.0	Static version	T-value	Reject (90% confidence)
MEAN: 1 STDEV: 1.05 n=19	MEAN: 0.50 STDEV: 1.09 n=18	-1.41	YES

Table 12: Result of T-test Performed on Improvement in User Perceived Proficiency

The result of the t-test indicates that the null hypothesis can be rejected with greater than a 90% degree of confidence; therefore, it can be concluded that over time the group interacting with ER-by-Design version 2.0 shows greater improvement in user perceived proficiency as compared to the group interacting with the static version of the system.

4.5 Summary

The null hypothesis states that a dynamic system that adapts itself to a user's current state of expertise in terms of the application domain, by constantly monitoring the user throughout their use of the system and placing them in appropriate user models when this expertise has changed, does not provide a more effective and efficient environment over time for users, when compared to a system with a static, generic interface. Relative improvement in amount of help usage, correctness scores for beginning level participants' final ER diagrams and user perception results of the system over time, did reject the null hypothesis with greater than a 90% degree of confidence. Relative improvement in task completion time, and complexity scores of the final ER diagrams did not reject the null hypothesis with greater than a 90% degree of confidence. These measurements of analysis are included in Table 13.

Measurement	Reject Null Hypothesis
Relative Improvement in Task Completion Time	NO
Relative Improvement in Amount of Help Usage	YES
Complexity of Final ER diagram	NO
Correctness of Final ER diagram (Beginning Level Group)	YES
User Perception of System Over Time	YES

Table 13: Measurements of Analysis

The conclusions drawn from these results will be discussed in chapter 5. Statistical results of all tests as well as data collected from Experiment 1 and Experiment 3 are included in Appendix G.

Chapter 5

CONCLUSIONS AND FUTURE RESEARCH

This research has investigated the enhancements, consisting of an inference engine, a help system, a help/assistance screen and user models, of a learning tool designed to lead the user through the creation of a visual Entity-Relationship diagram. The goals of the research were to examine the effects over time of a dynamic system that incorporated user models and adapted itself to a user's current state of expertise, in terms of the application domain, by placing the user in the appropriate model when the system had inferred that the user's expertise had changed.

To evaluate these effects, ER-by-Design version 2.0 was developed and compared to a generic, static version of the system, that did not consist of the enhancements, therefore not incorporating the concepts of user models and interface adaptation. Evaluation consisted of analyzing data gathered from Experiments 1, 2 and 3 pertaining to the relative improvement in task completion time, relative improvement in amount of help usage, complexity and correctness of the final diagrams created with the system and user perception of the system over time.

5.1 Conclusions

Although not significant, statistical analysis of data gathered indicates that over time graduate student participants interacting with ER-by-Design version 2.0 displayed the most relative improvement in task completion time as compared to graduate students interacting with the static version of the system. Over time participants with less experience and understanding of ER design concepts interacting with ER-by-Design version 2.0 did not show such a relative improvement in task completion time as compared to participants with less experience interacting with the static version of the system. It appears that participants who had a certain level of mastery of the concepts relating to ER design benefited the most from ER-by-Design version 2.0, in terms of relative improvement in task completion time, while the participants who were only beginning to understand the concepts of ER design, did not. Several reasons could contribute to this. Because this information would be new to them, participants with less experience and understanding of ER modeling concepts may have spent more time processing the information presented on the help/assistance screens. In addition, participants may not have had a sufficient time frame in order to show a significant relative improvement in task completion time. Participants unfamiliar with ER modeling concepts may have benefited from more prolonged use of the system.

These results are similar to Zellermayer's results of his study of the Writing Partner system. He found that students who were cued with unsolicited advice took longer to write their essays and did not show initial improvement; however, two weeks later, the advised students wrote better essays as compared to the students who had interacted with the system that did not give unsolicited advice. [Zellermayer91]

Similar results to Zellermayer's were found in this study of ER-by-Design version 2.0, considering the relationship between relative improvement in task completion time and correctness of the final ER diagrams. Similarly to Zellermayer's study, analysis of the data indicates that beginning level participants interacting with ER-by-Design version 2.0, while not displaying significant relative improvement in task completion time, created more correct final ER diagrams in Experiment 3, as compared to participants interacting with the static version of the system. Participants in the beginning level group possessed the least domain knowledge, and had the least prior experience with the concepts of ER modeling; therefore, they would be the most likely to benefit from the system in terms of correctness, or creating better ER diagrams.

In addition, analysis of data indicates that participants interacting with ER-by-Design version 2.0 displayed significant relative improvement in amount of help usage when compared to participants interacting with the static version of the system. Help usage for the adaptive system decreased approximately 50 percent from the first to the last experiment while help usage for the static version only decreased approximately 27 percent from the first to the last experiment. Furthermore, for the third and final experiment, the number of times help was accessed by participants interacting with ER-by-Design version 2.0 was 91 percent less than the number of times help was accessed by participants interacting with the static version of the system.

Data gathered did not indicate a significant difference in terms of the complexity of the final diagrams created in Experiment 3 by both systems. This lack of significance in terms of complexity may have been a result of constraints of the experiment that required all participants to create an ER diagram of the same scope, not allowing for much diversity between the ER diagrams created. Despite this constraint, the graduate participants still created ER diagrams approximately 18 percent more complex with ER-by-Design version 2.0 than with the static version of the system.

Lastly, data gathered indicates a significant difference in user perception over time for ER-by-Design version 2.0 when compared to the static version of the system. For Experiment 3, perception of the adaptive system ER-by-Design version 2.0 was approximately 7 percent more positive than perception of the static version of the system.

In terms of Questions 1 through 7 of the user survey for the last experiment, user perception for the adaptive system was significantly more positive than user perception for the static version of the system for Questions 4, 5 and 6. Questions 1, 2, 3 and 7 did not show such a significant difference.

Question 4 collected data on the level of frustration related to the completion of the task; Question 5 collected data on the participant's ability to complete the task quickly with the system and Question 6 collected data on how comfortable the participant was with the interface of the system. The results suggest that the participants interacting with ER-by-Design version 2.0 felt more comfortable with the interface of the system, were able to complete tasks quicker and were less frustrated than participants interacting with the static version of the system for the final experiment.

Question 1 collected data on the ease of use of the system; Question 2 collected data on how successful the participant felt they were in completing the task; Question 3 collected data on how helpful the system was for the task and Question 7 collected data on the level of assistance's appropriateness to the participant's level of domain knowledge. The results do not suggest that the participants interacting with ER-by-Design version 2.0 felt the system was easier to use, led to more success in task completion, was more helpful and was appropriate to the participant's domain knowledge than participants interacting with the static version of the system.

Significant differences between the two systems for Questions 1, 2, 3 and 7 may not have been reflected because the participants interacting with ER-by-Design did not feel ready when they were advanced to more advanced user models. They may have become dependent of the level of help in the naive and beginning user models whereas the participant's interacting with the static version of the system did not have such concentrated help.

Data gathered also indicates that participants interacting with ER-by-Design version 2.0 showed greater improvement in their self-ranking scores than participants interacting with the static version of the system. On a ranking scale of 1 to 7, participants interacting with ER-by-Design felt their proficiency in the domain of ER design had improved by an approximate average of 14 percent throughout the experiments while participants interacting with the static version of the system felt their proficiency in the domain of ER design had improved only by an approximate average of 7 percent throughout the experiments.

Data gathered over time also indicates that user perception of ER-by-Design version 2.0 improved by 21 percent while user perception of the static version of the system decreased by 52 percent. Overall, it appears that participants interacting with the adaptive system, ER-by-Design version 2.0, felt better about the system the longer they used it, while participants interacting with the static version of the system appreciated the system initially, however, became frustrated and felt less positive about it over time.

Overall, the results of the analysis of the data gathered indicates that over time, ER-by-Design version 2.0 provides a more efficient and effective environment than the static version of the system that does not adapt dynamically to the user based on the user's domain expertise. For undergraduate users, this increase in efficiency and effectiveness was not as obvious and could be a result of the learning process. The adaptive system was saturated with information related to ER design concepts and may have overwhelmed participants who were just beginning to grasp the concepts and were consequently having to concentrate and absorb such a large amount of information, as opposed to those who had more knowledge and experience with such concepts. It is important to note that ER-by-Design version 2.0, although designed as a learning tool, was not designed as a standalone application, and was designed not to teach ER design concepts but to aid the learning of these concepts as an enhancement to the classroom environment.

5.2 Future Research

Several issues for future work and exploration are suggested by this research. Among these are additional metrics for analysis of results and adaptation of the system as well as improvements in the inference engine and applying the concept of the user models.

During the first experiment, participants were allowed to ask questions pertaining to the domain of ER design and to use their textbooks. It was observed that participants interacting with the static version of the system asked many more questions and also used their textbooks much more often than participants interacting with ER-by-Design version 2.0 did. The behavior of the participant, relating to these observations, would have been an interesting and quite useful metric in terms of analyzing the usefulness of the adaptive version of the system.

Another area for improvement concerns accuracy data. Accuracy data is most important in judging a user's expertise and one of the most difficult metrics to apply for a system with a domain such as ER concepts. A system that determines the expertise of the user by analyzing the accuracy of the data created with the system would be most useful. An enhancement to ER-by-Design version 2.0 could be an accuracy metric that uses the steps of an algorithm for ER-to-Relational mapping. Unfortunately, the work involved with implementing such a metric for ER-by-Design version 2.0 was out of the scope of this research; however, the accuracy of the system could greatly benefit from such an enhancement.

Another important enhancement would be the ability of the inference component to not only determine an increase in accuracy but a decrease as well. When a significant decrease in expertise has been determined, the user would then be put in a user model more appropriate for this new, and reduced, level of expertise. This enhancement would be most appropriate for users who have not used the system for a while in addition to not practicing ER design concepts and who may experience a different level of understanding until they have the opportunity to "relearn" what was forgotten. Intermediate users would probably benefit the most from such an enhancement.

Lastly, an additional enhancement that would benefit the system would be ability of the system to place each user in an appropriate model at the beginning of the user's interaction with the system, instead of forcing the user to begin in the naive model and move through each model in the order of the levels of naive, beginner, intermediate and advanced. This constraint of the existing system can be quite frustrating for a user with a high level of expertise who is forced to work through the naive, beginner and intermediate user models for at least eight sessions until advancement to the advanced user model has been reached. Such an enhancement may entail an inference component that is able to determine user expertise dynamically without having to analyze the data only after every session. Furthermore, it may also be possible for the user to complete a brief user survey pertaining to their previous experience with ER modeling concepts and for this information to be passed on to the inference component for analysis. Several more possibilities may exist for creating such functionality in an adaptive system.

These possibilities present opportunities to continue and expand the study of user models in a dynamically adaptive system. This research has contributed to the groundwork for such future research.

REFERENCES

[Barnett98]
Barnett, L. J., et al., "Design and Implementation of an Interactive Tutorial Framework," Proceedings of the Twenty-Ninth SIGCSE Technical Symposium on Computer Science Education (1998), pp. 87-91.

[Beck97]
Beck, J., M. Stern and B. P. Woolf, "Using the Student Model to Control Program Difficulty [sic]," User Modeling: Proceedings of the Sixth International Conference (1997), pp. 277-288.

[Belge96]
Belge, M. and K. Ehrlich, "The User Model as a Discipline for Interface Design," CHI '96. Proceedings of the CHI '96 Conference Companion on Human Factors in Computing Systems: Common Ground (1996), p. 421.

[Benyon93]
Benyon, D. and D. Murray, "Developing Adaptive Systems to Fit Individual Aptitudes," Proceedings of the 1993 ACM International Workshop on Intelligent User Interfaces (1993), pp. 115-121.

[Birnbaum97]
Birnbaum, L., et al., "Compelling Intelligent User Interfaces - How Much AI?," Proceedings of the 1997 ACM International Workshop on Intelligent User Interfaces (1997), pp. 173-175.

[Bonar91]
Bonar, J. and B. Liffick, "Communicating with High-Level Plans," Intelligent User Interfaces, J. W. Sullivan and S. W. Tyler, eds., ACM Press, NewYork, 1991, pp. 129-156.

[Cohen98]
Cohen, P. R., et al., "Synergistic Use of Direct Manipulation and Natural Language," Readings In Intelligent User Interfaces, M. T. Maybury and W. Wahlster, eds., Morgan Kaufmann Publishers, Inc., San Francisco, 1998, pp. 29-35.

[Crow93]
Crow, D. and B. Smith, "The Role of Built-in Knowledge in Adaptive Interface Systems," Proceedings of the 1993 ACM International Workshop on Intelligent User Interfaces (1993), pp. 97-104.

[Desmarais93]
Desmarais, M. C. and J. Liu, "Exploring the Applications of User-Expertise Assessment for Intelligent Interfaces," CHI '93, Conference Proceedings on Human Factors in Computing Systems (1993), pp. 308-313.

[Dicks94]
Dicks, R. S., "Integrating Online Help, Documentation, and Training," Proceedings of the 1994 ACM Conference on Technical Communications at the Great Divide (1994), pp. 115-118.

[Dryer97]
Dryer, D. C., "Wizards, Guides, and Beyond: Rational and Empirical Methods for Selecting Optimal Intelligent User Interface Agents," Proceedings of the 1997 ACM International Conference on Intelligent User Interfaces (1997), pp. 265-268.

[Elmasari94]
Elmasari, R. and S. Navathe, Fundamentals of Database Systems Second Edition, Addison-Wesley Publishing Company, Menlo Park, California, 1994, pp. 39-68.

[Kass91]
Kass, R., "The Ideal General User Model," Intelligent User Interfaces, J. W. Sullivan and S. W. Tyler, eds., ACM Press, New York, 1991, pp. 111-128.

[Kelly98]
Kelly, C. L., et al., "Reliability Assessment of a User Proficiency Measurement Technique," International Journal of Human Computer Interaction 10, 1 (1998), pp. 33-49.

[Knabe95]
Knabe, K., "Apple Guide: A Case Study in User-Aided Design of Online Help," CHI '95, Conference Companion on Human Factors in Computing Systems (May, 1995), pp. 286-287.

[Maybury98]
Maybury, M. T. and W. Wahlster, eds., Readings in Intelligent User Interfaces, M. T. Maybury and W. Wahlster, eds., Morgan Kaufmann Publishers, Inc., San Francisco, 1998.

[Meyer93]
Meyer, B., K. C. B. Yakemovie and S. W. Tyler, "Issues In Practical Application Of An Adaptive Interface," Proceedings of the 1993 ACM International Workshop on Intelligent User Interfaces (1993), pp. 251-254.

[Miller91]
Miller, J. R., et al., "Introduction," Intelligent User Interfaces, Sullivan, J. W. and S. W. Tyler, eds., ACM Press, NewYork, 1991, pp. 1-10.

[Murphy97]
Murphy, M. and M. McTear, "Learner Modeling for Intelligent CALL," User Modeling: Proceedings of the Sixth International Conference (1997), pp. 301-312.

[Oppermann94]
Oppermann, R., ed., Adaptive User Support Ergonomic Design of Manually and Automatically Adaptable Software, Lawrence Erlbaum Associates, New Jersey, 1994.

[Pangoli95]
Pangoli, S. and F. Paterno, "Automatic Generation of Task-oriented Help," Proceedings of the 1995 ACM Symposium on User Interface and Software Technology (November, 1995), pp. 181-187.

[Paris89]
Paris, C., "The Use of Explicit User Models in a Generation System for Tailoring Answers to the User's Level of Expertise," User Models in Dialog Systems, A. Kobsa, et al., eds., Springer-Verlag, New York, 1989, pp. 200-232.

[Rich79]
Rich, E., "User Modeling via Stereotypes," Readings In Intelligent User Interfaces, M. T. Maybury and W. Wahlster, eds., Morgan Kaufmann Publishers, Inc., San Francisco, 1998, pp. 329-341.

[Rich83]
Rich, E., "Users are individuals: individualizing user models," International Journal of Man-Machine Studies 18 (1983), pp. 199-214.

[Schlungbaum97]
Schlungbaum, E., "Individual User Interfaces and Model-based User Interface Software Tools," Proceedings of the 1997 ACM International Workshop on Intelligent User Interfaces (1997), pp. 229-232.

[Selker94]
Selker, T., "Coach: A Teaching Agent that Learns," Communications of the ACM 37, 7 (July 1994), pp. 92-99.

[Sukaviriya93]
Sukaviriya, P. N. and J. D. Foley, "Supporting Adaptive Interfaces in a Knowledge-Based User Interface Environment," Proceedings of the 1993 ACM International Workshop on Intelligent User Interfaces (1993), pp. 107-113.

[Sullivan91]
Sullivan, J. W. and S. W. Tyler, eds., Intelligent User Interfaces, ACM Press, New York, 1991.

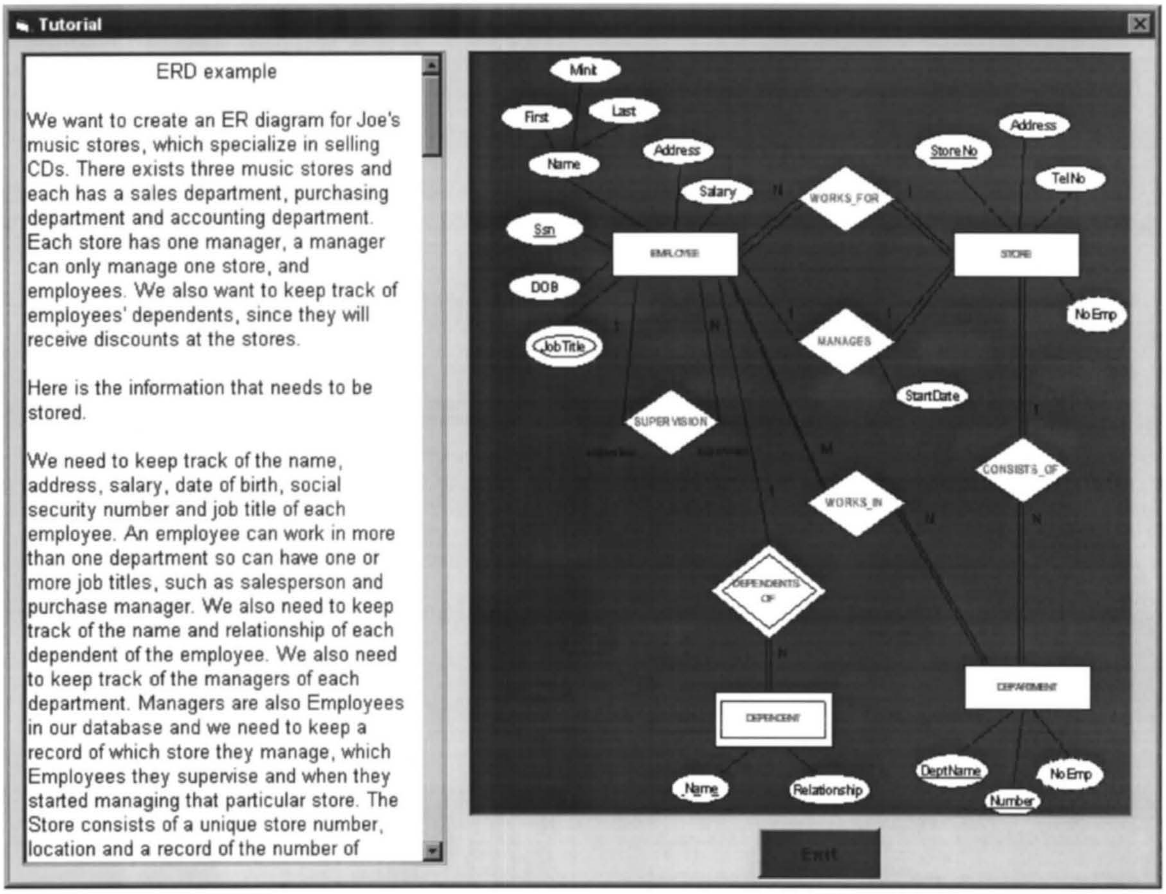
[Tyler91]
Tyler, S. W., et al., "An Intelligent Interface Architecture for Adaptive Interaction," Intelligent User Interfaces, J. W. Sullivan and S. W. Tyler, eds., ACM Press, New York, 1991, pp. 85-109.

[Wahlster89]
Wahlster, W. and A. Kobsa, "User Models in Dialog Systems," User Models in Dialog Systems, A. Kobsa, et al., eds., Springer-Verlag, New York, 1989, pp. 4-18.

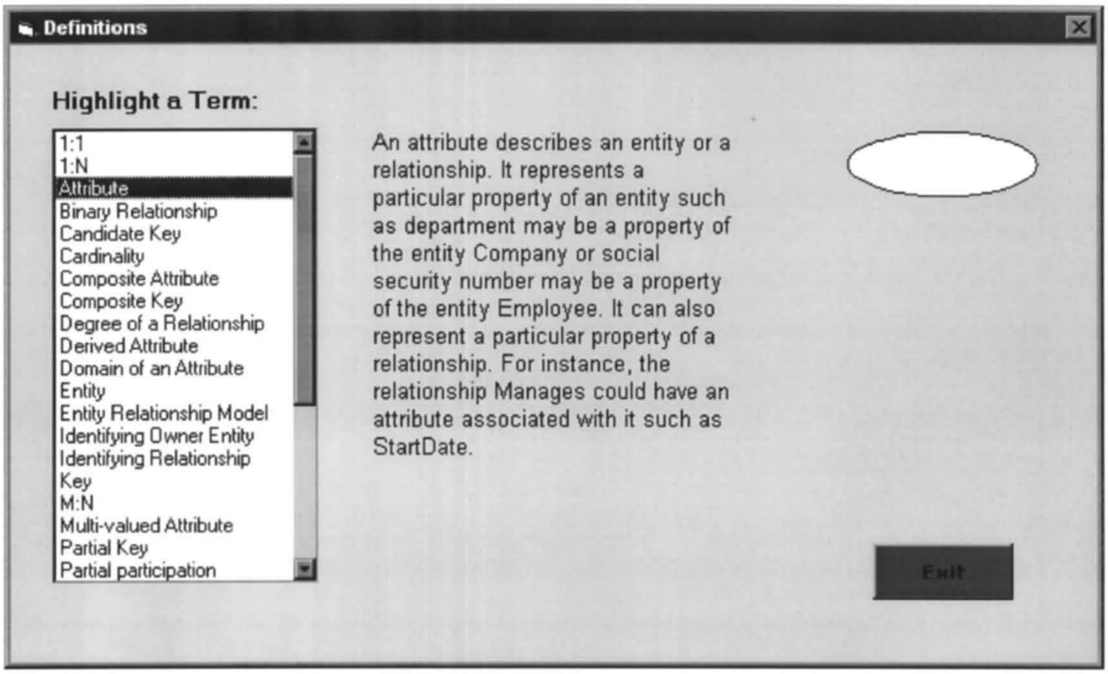
[Zellermayer91]
Zellermayer, M., et al., "Enhancing Writing-Related Metacognitions Through a Computerized Writing Partner," American Educational Research Journal 28, 2 (Summer, 1991), pp. 373-391.

APPENDIX A

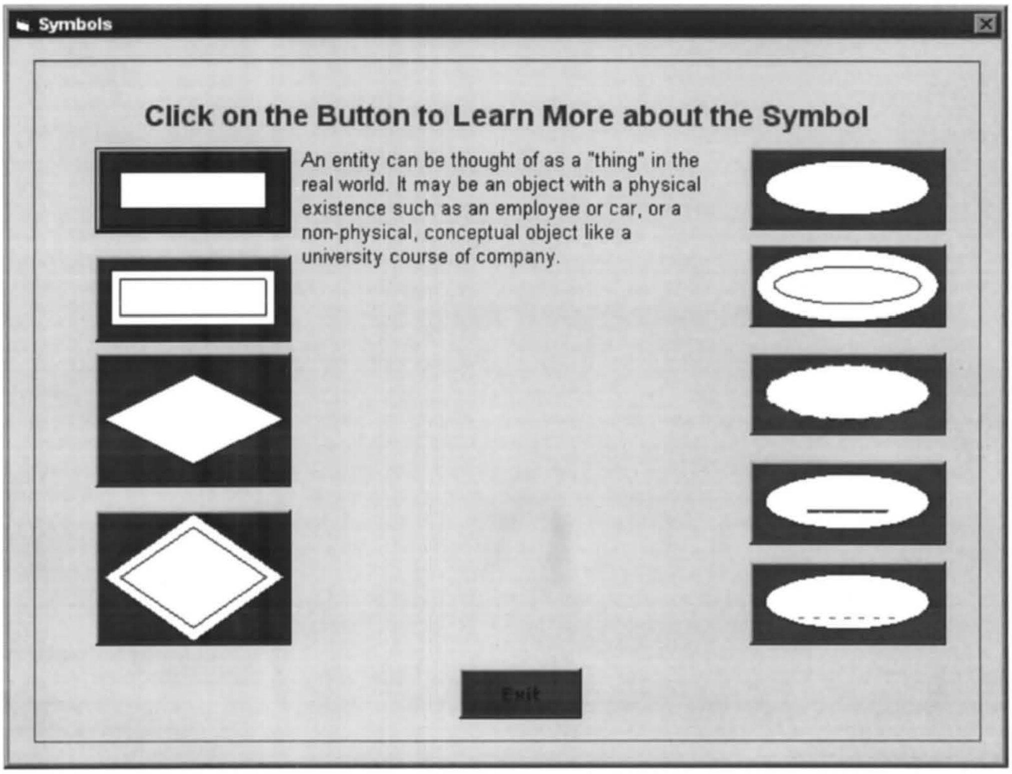
Help Modules Created for ER-by-Design Version 2.0


Tutorial
ERD example
We want to create an ER diagram for Joe's music stores, which specialize in selling CDs. There exists three music stores and each has a sales department, purchasing department and accounting department. Each store has one manager, a manager can only manage one store, and employees. We also want to keep track of employees' dependents, since they will receive discounts at the stores.
Here is the information that needs to be stored.
We need to keep track of the name, address, salary, date of birth, social security number and job title of each employee. An employee can work in more than one department so can have one or more job titles, such as salesperson and purchase manager. We also need to keep track of the name and relationship of each dependent of the employee. We also need to keep track of the managers of each department. Managers are also Employees in our database and we need to keep a record of which store they manage, which Employees they supervise and when they started managing that particular store. The Store consists of a unique store number, location and a record of the number of
Mnit
First
Last
Name
Address
Salary
Ssn
DOB
JobTitle
WORKS_FOR
MANAGES
StoreNo
Address
TelNo
NoEmp
StartDate
SUPERVISION
CONSISTS_OF
WORKS_IN
DEPENDENTS OF
Name
Relationship
DeptName
Number
NoEmp
Exit

Terminology Module



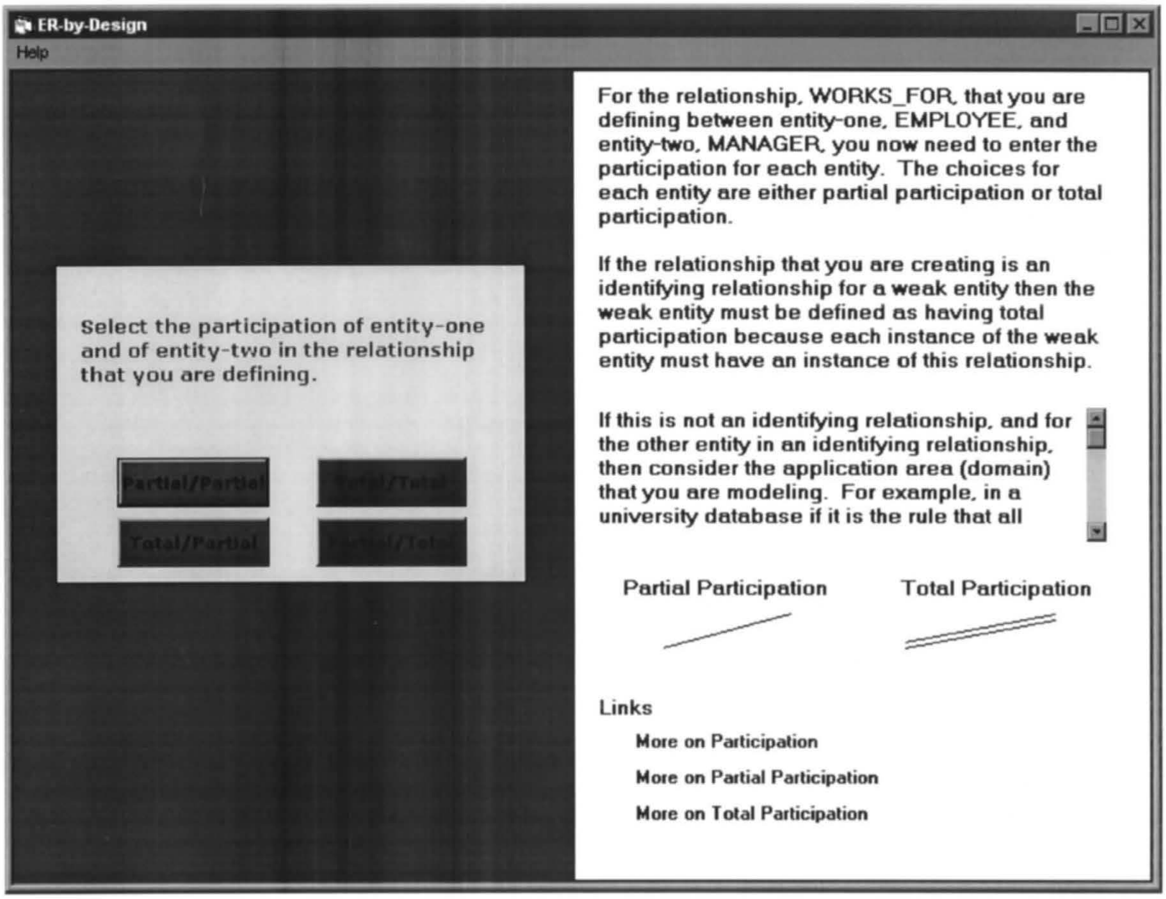
Symbol Lookup Module



APPENDIX B

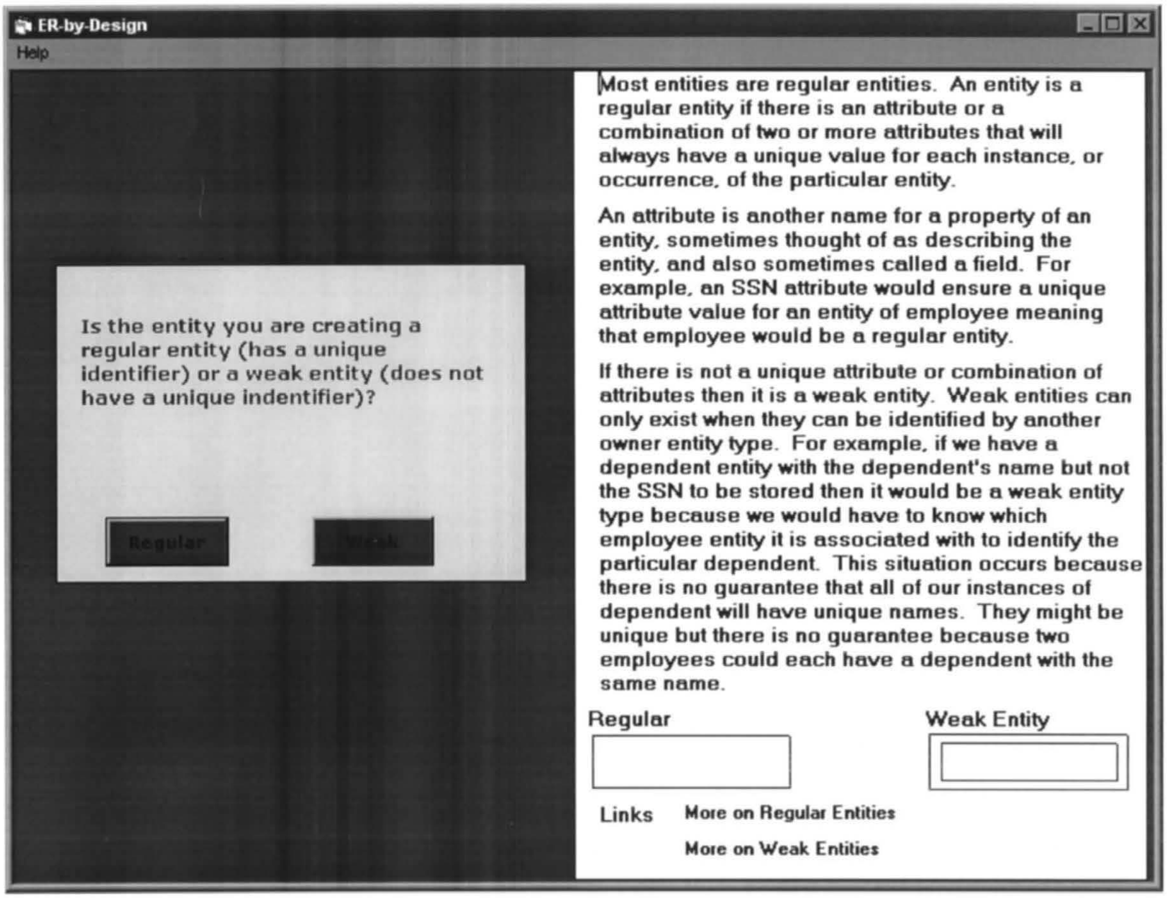
Help/Assistance Screen

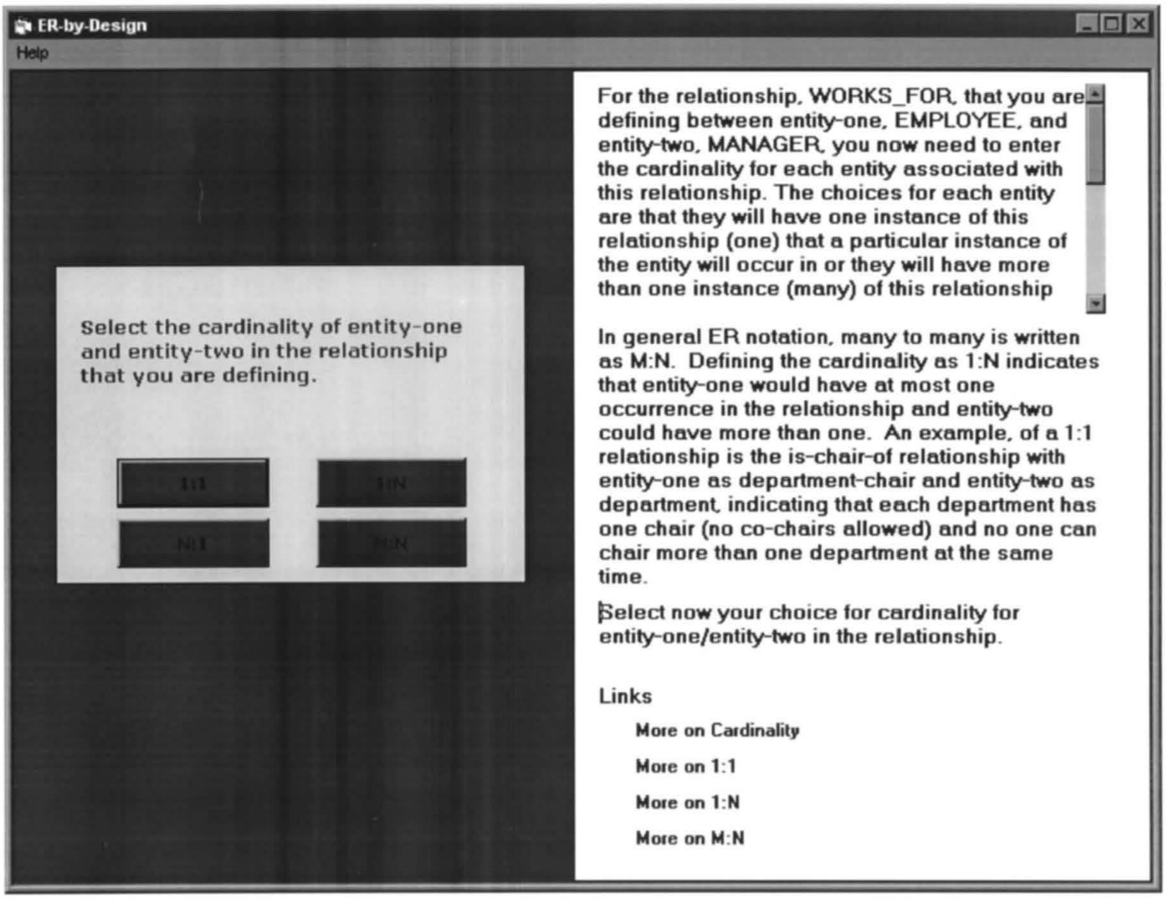
Help/Assistance Screen



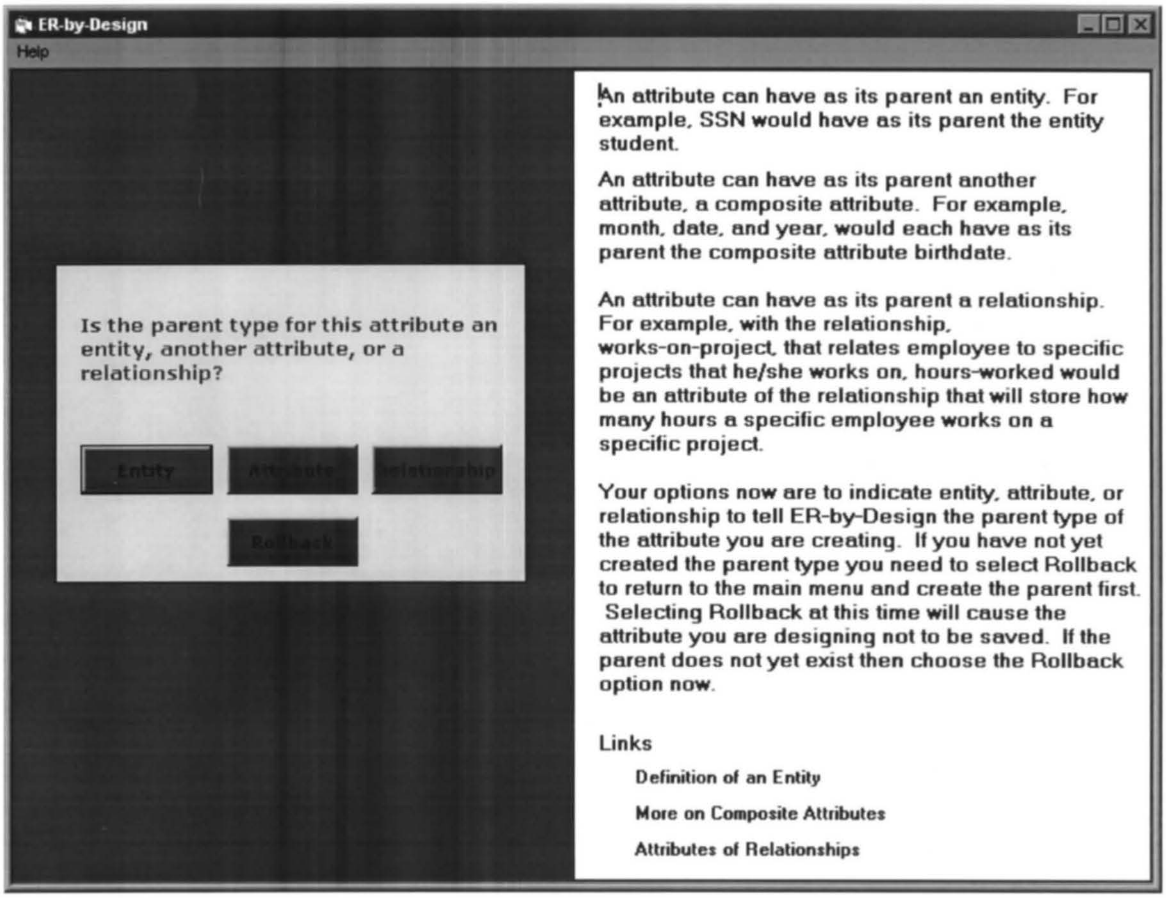
APPENDIX C

User Models

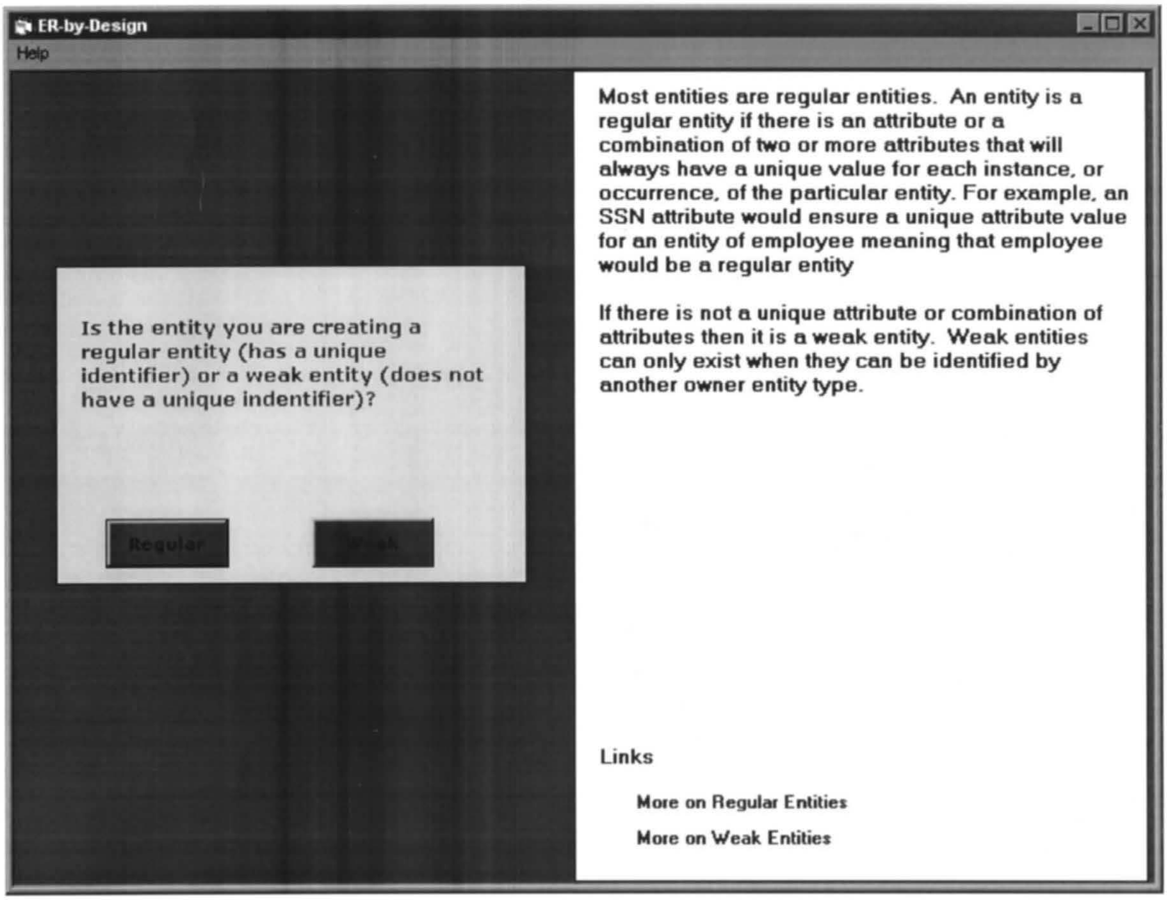

ER-by-Design
Help
Is the entity you are creating a regular entity (has a unique identifier) or a weak entity (does not have a unique indentifier)?
Most entities are regular entities. An entity is a regular entity if there is an attribute or a combination of two or more attributes that will always have a unique value for each instance, or occurrence, of the particular entity.
An attribute is another name for a property of an entity, sometimes thought of as describing the entity, and also sometimes called a field. For example, an SSN attribute would ensure a unique attribute value for an entity of employee meaning that employee would be a regular entity.
If there is not a unique attribute or combination of attributes then it is a weak entity. Weak entities can only exist when they can be identified by another owner entity type. For example, if we have a dependent entity with the dependent's name but not the SSN to be stored then it would be a weak entity type because we would have to know which employee entity it is associated with to identify the particular dependent. This situation occurs because there is no guarantee that all of our instances of dependent will have unique names. They might be unique but there is no guarantee because two employees could each have a dependent with the same name.
Regular
Weak Entity
Links
More on Regular Entities
More on Weak Entities


ER-by-Design
Help
Select the cardinality of entity-one and entity-two in the relationship that you are defining.
For the relationship, WORKS_FOR, that you are defining between entity-one, EMPLOYEE, and entity-two, MANAGER, you now need to enter the cardinality for each entity associated with this relationship. The choices for each entity are that they will have one instance of this relationship (one) that a particular instance of the entity will occur in or they will have more than one instance (many) of this relationship
In general ER notation, many to many is written as M:N. Defining the cardinality as 1:N indicates that entity-one would have at most one occurrence in the relationship and entity-two could have more than one. An example, of a 1:1 relationship is the is-chair-of relationship with entity-one as department-chair and entity-two as department, indicating that each department has one chair (no co-chairs allowed) and no one can chair more than one department at the same time.
Select now your choice for cardinality for entity-one/entity-two in the relationship.
Links
More on Cardinality
More on 1:1
More on 1:N
More on M:N

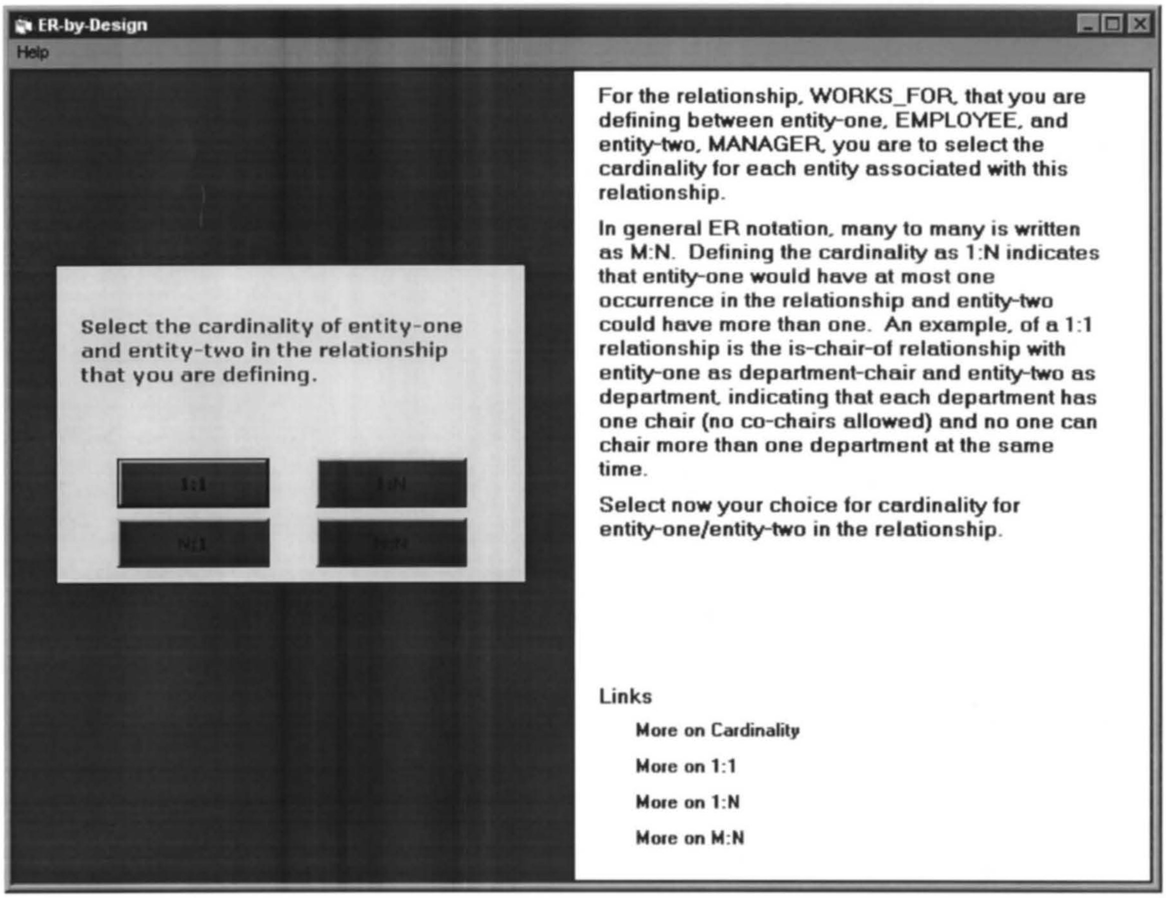
Beginning User Module - Creating an Attribute



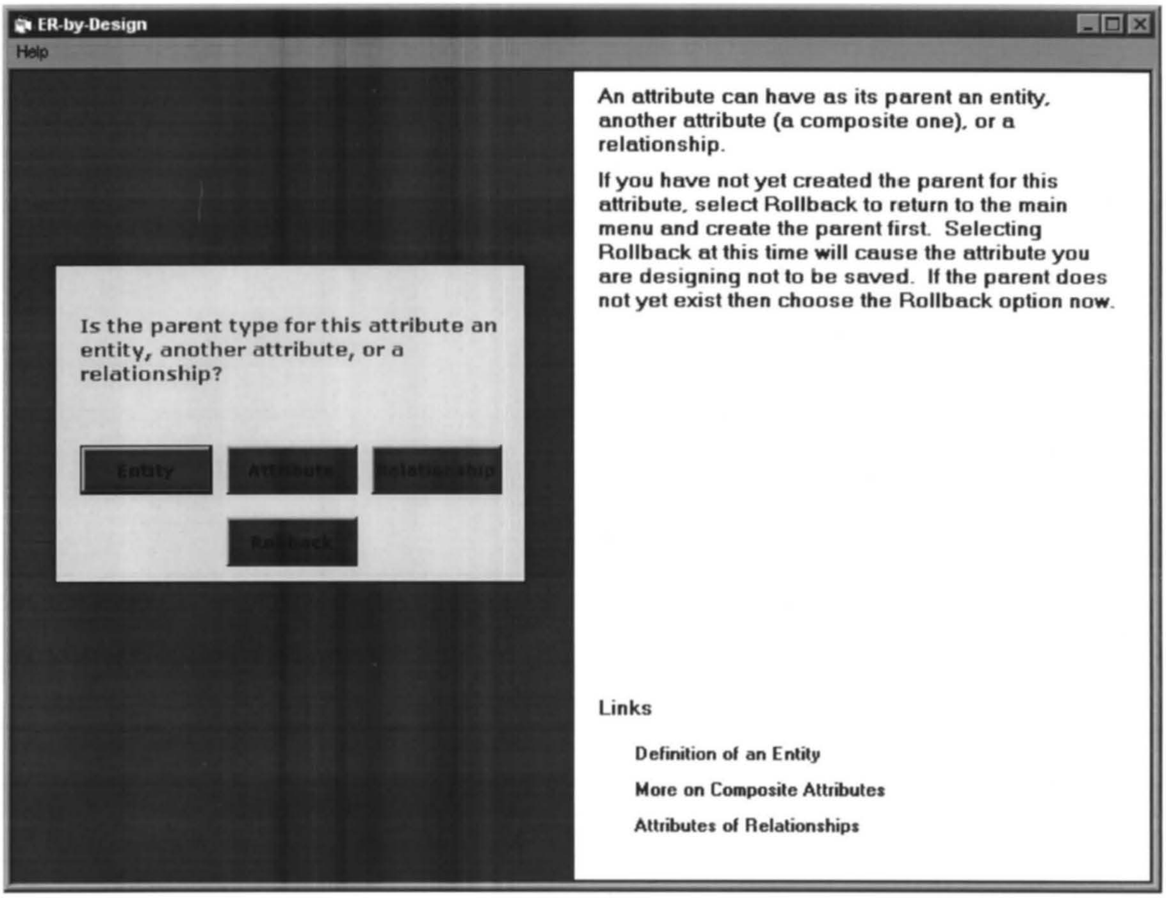
Intermediate User Model - Creating an Entity



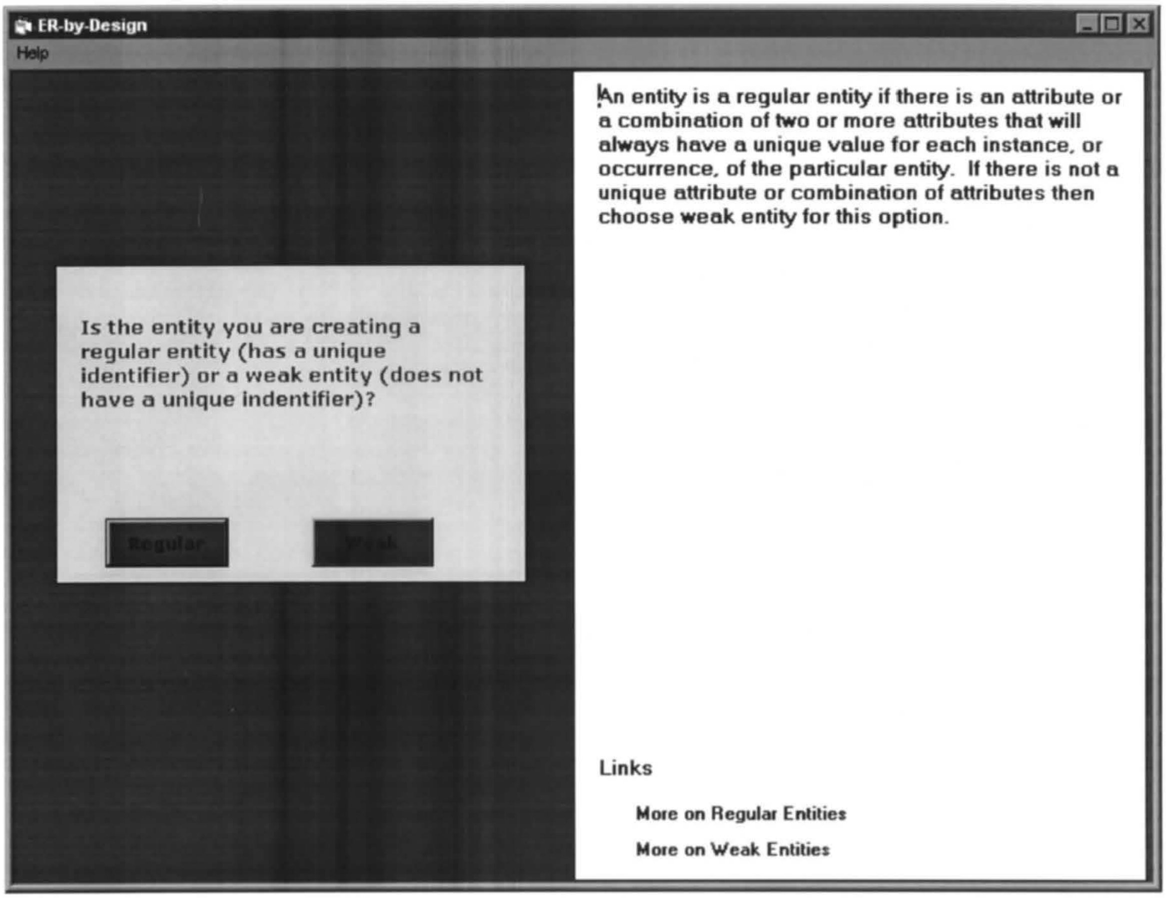
Intermediate User Model - Creating a Relationship



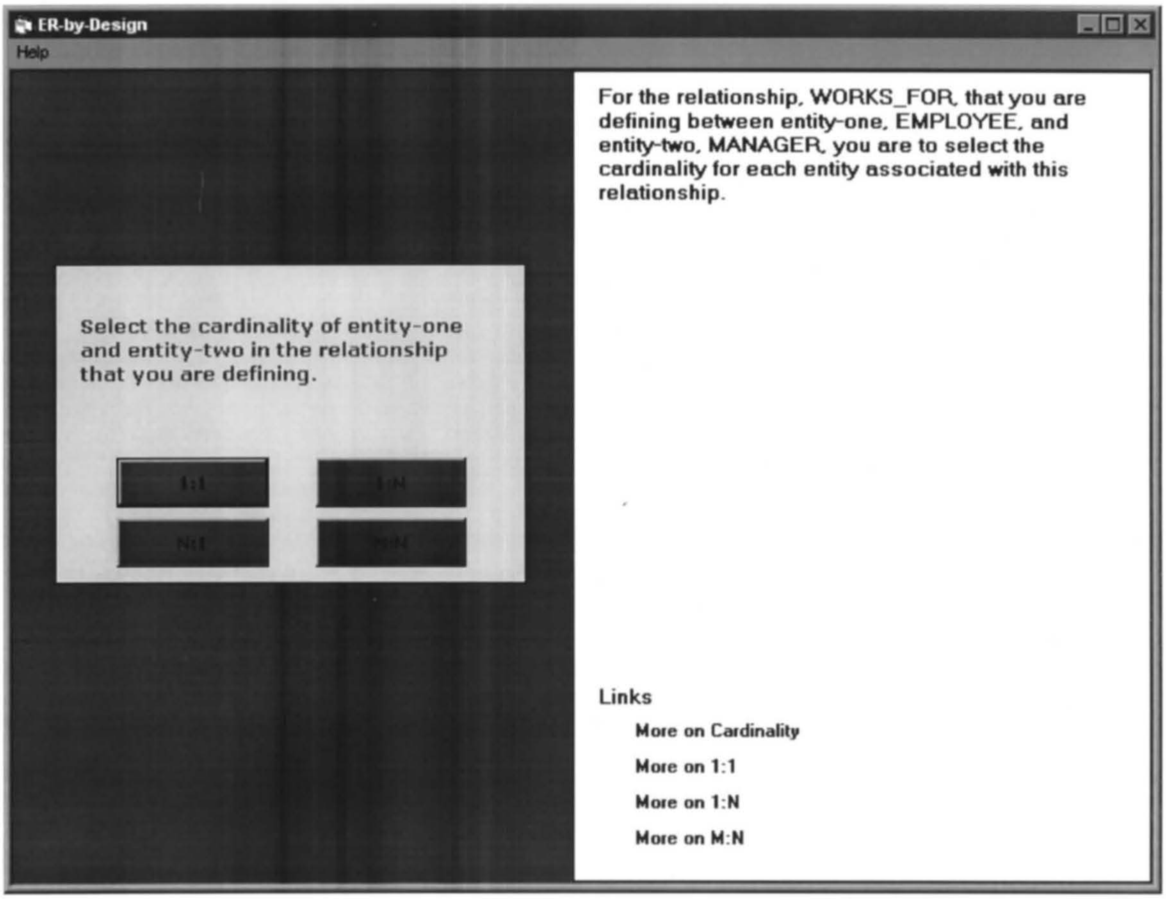
Intermediate User Model - Creating an Attribute



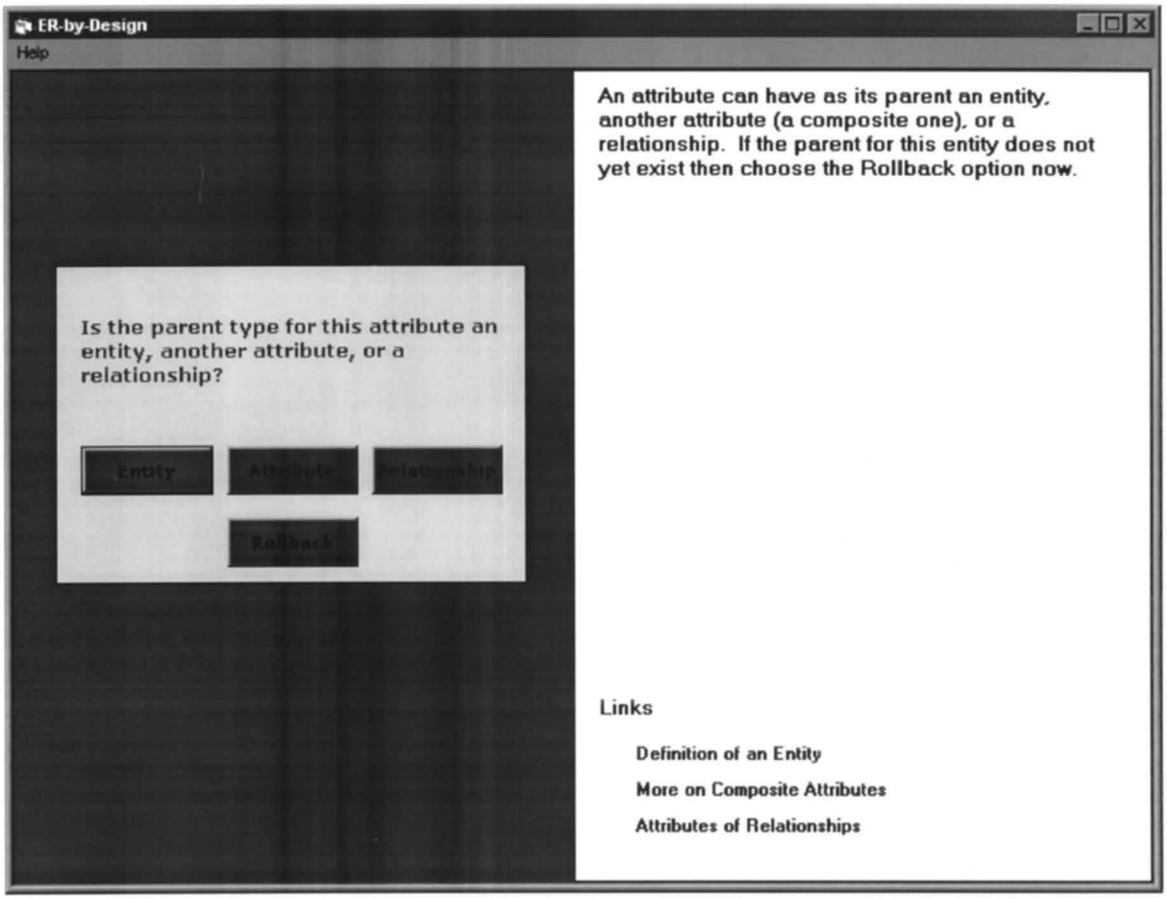
Advanced User Model - Creating an Entity



Advanced User Model - Creating a Relationship



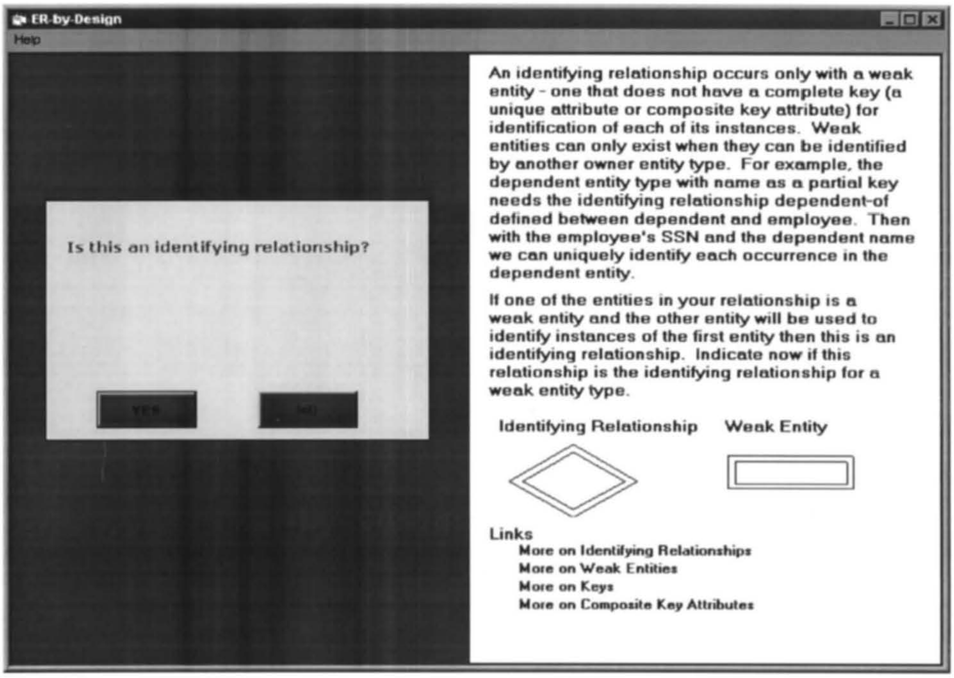
Advanced User Model - Creating an Attribute



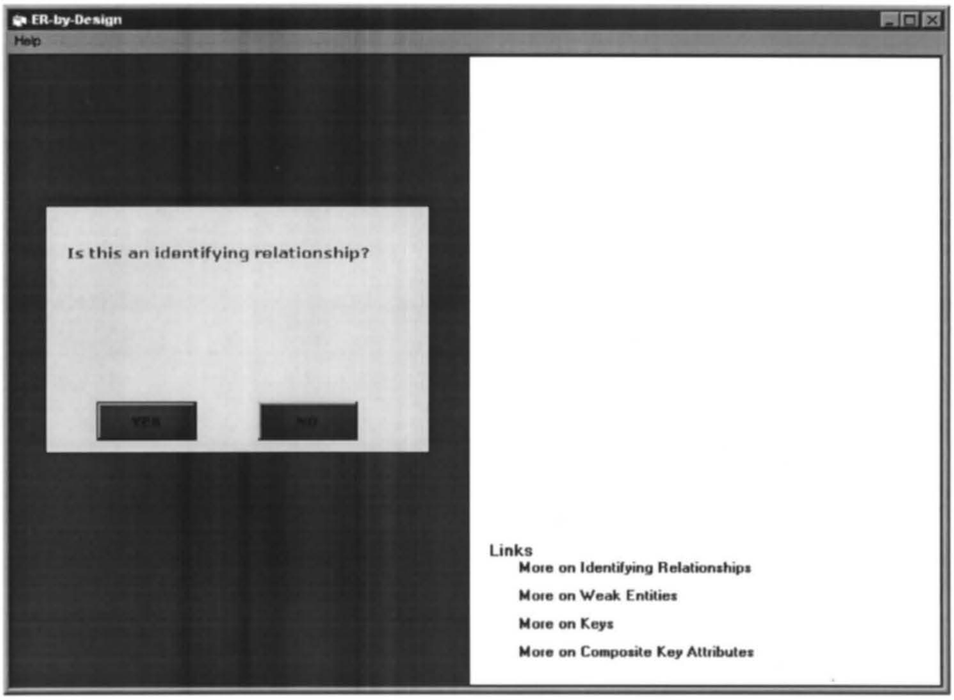
APPENDIX D

Interfaces for ER-by-Design Version 2.0 and Static Version

ER-by-Design Version 2.0



Static Version



APPENDIX E

Exercises for Experiments 1, 2 and 3

Experiment 1

I. You are to create a very simple ER diagram using ER_by_Design to have two entities and one relationship as follows:

First entity: employee with attributes - SSN (a key field), emp-name, address

Second entity: department with attributes - code (a key field), dept-name (a key field), location (multi-valued)

Relationship: assigned-to - a one-to-many (1:N) relationship between department and employee with total participation for both entities

After you have created the two entities, the relationship and the six attributes, use the display entity and display relationship functions on the main menu to see a description of what you have created. Then exit and save your diagram to emp-dept.dat.

II. You are to create a new ER diagram to store information about students and courses. For students, we want to store their SSN (unique), name, address, and phone. For courses, we want to store a code (unique), name, and description. We also want to associate students with the courses they have enrolled in. Assume that we will store students in our database prior to them enrolling in their first course and we want to store information about our courses prior to the first time they are offered.

After you have created the two entities, the relationship, and the attributes, use the display entity and display relationship functions on the main menu to see a description of what you have created. Then exit and save your diagram to student.dat.

III. Begin execution of ER_by_Design again and this time choose to open a file. When given a choice, select and open student.dat.

First, use the display functions to see that the information you stored during the previous execution has been loaded. Now, select the draw function and experiment with how this works.

Then exit and do not save your diagram. (The original file will still be on your disk.)

IV. Begin a new execution of ER_by_Design and this time choose to begin a new ER diagram. You are to store data about a library. You are to store information about its patrons to include an id, a name, an address, and a telephone number. We might want to refer to the entire address of a patron or we might want to find out information such as all the patrons from a specific city. You are to store information about each book to include its ISBN, its name, and the shelf location where it is stored. You are to store publisher information to include a code, publisher name, address, and a contact name. We would like to keep information to include the books that a patron has borrowed. We would also like to know the publisher for each of our books.

Design an ER using ER_by_Design then save your work as library.dat.

Begin execution again and open library.dat. Display the entity and relationship info to see that it is correct then use the draw routine to create a drawing of the entities and relationships.

Experiment 2

You are to design an ER diagram to model the following information. Use the notation as in Figure 3.2 in the Elmasri and Navathe text. Include any additional assumptions you made in developing the conceptual model.

A database is being designed to keep track of NFL teams and games. A team has many players, not all participating in each game. We wish to keep track of the players for each game for each team, the positions played in that game, who won, and the score. For each team we wish to keep its name; location (city); address; home stadium; box office phone number; owner with address and phone number; coach with address, phone number, and current salary; season record; and overall record. For each player we wish to keep his name; start date with the team; start date with the league; prior team affiliations; years played; positions played; and current salary. For each game we wish to keep the names of the two teams that played, the date, the time, location, score, and winner.

You are to prepare this assignment individually. To be submitted: a list of any additional assumptions that you made, a printout showing the entities and the relationships using ER_by_Design, a disk with your saved data file for your ER schema.

Experiment 3

You are to design an ER diagram to model the following information. A database is being designed to keep track of data related to automobile insurance policies for an insurance company. You wish to record information about the following.

Agents - id, name (first and last), address (street, city, state, zip), phone(s)

Buyers - SSN, name (first and last), address (street, city, state, zip)

Vehicles - VIN, make, model, year

Additionally, for each policy we wish to store the policy # (unique), the associated agent(s), the associated buyer, the associated vehicle, the cost of the vehicle, the cost of the policy, and the date the policy becomes effective.

For this database, we will assume that the following rules apply. More than one agent may be associated with a policy. We will only store one buyer name associated with each vehicle. We will only have one policy associated with a vehicle.

Design an ER schema that models exactly the preceding information.

APPENDIX F

User Surveys

User Survey for ER-by-Design version 2.0

Please use the following scale:

1. Strongly Disagree
2. Disagree
3. Somewhat Disagree
4. No Opinion
5. Somewhat Agree
6. Agree
7. Strongly Agree

Please circle your response.

1.) This system was very easy to use.
1 2 3 4 5 6 7

2.) I was successful in completing my task.
1 2 3 4 5 6 7

3.) This system was very helpful for this task.
1 2 3 4 5 6 7

4.) I was able to complete my task with a minimum level of frustration.
1 2 3 4 5 6 7

5.) I was able to complete my task quickly.
1 2 3 4 5 6 7

6.) I was comfortable with the interface of the program.
1 2 3 4 5 6 7

7.) The system's level of assistance was appropriate for my level of knowledge of creating ER diagrams
1 2 3 4 5 6 7

8.) Rate your level of expertise in designing ER diagrams.
1 2 3 4 5 6 7
beginner intermediate advanced

9.) Please feel free to list any comments or suggestions you may have on the back of this survey. For instance, were you comfortable as the changes in the interface occurred? Did the interface allow you to complete your task more effectively and efficiently?

User Survey for Static Version

Please use the following scale:

1. Strongly Disagree
2. Disagree
3. Somewhat Disagree
4. No Opinion
5. Somewhat Agree
6. Agree
7. Strongly Agree

Please circle your response.

1.) This system was very easy to use.
1 2 3 4 5 6 7

2.) I was successful in completing my task.
1 2 3 4 5 6 7

3.) This system was very helpful for this task.
1 2 3 4 5 6 7

4.) I was able to complete my task with a minimum level of frustration.
1 2 3 4 5 6 7

5.) I was able to complete my task quickly.
1 2 3 4 5 6 7

6.) I was comfortable with the interface of the program.
1 2 3 4 5 6 7

7.) The system's level of assistance was appropriate for my level of knowledge of creating ER diagrams
1 2 3 4 5 6 7

8.) Rate your level of expertise in designing ER diagrams.
1 2 3 4 5 6 7
beginner intermediate advanced

9.) Please feel free to list any comments or suggestions you may have on the back of this survey. For instance, did you wish the system offered more help and assistance based of your level of knowledge in the creation of ER diagrams?

APPENDIX G

Statistical Results

Relative Improvement in Task Completion Time

Average Task Completion Time

Static version	**ER-by-Design version 2.0**	**T-value**	**Reject (90% confidence)**
MEAN:98.35 STDEV:36.42 n=18	MEAN:86.62 STDEV:31.48 n=19	1.04	NO (85%)

Average Task Completion Time for Experiment 1

Static version	**ER-by-Design version 2.0**	**T-value**	**Reject (90% confidence)**
MEAN:136.43 STDEV:47.84 n=18	MEAN:114.65 STDEV:52.59 n=19	1.31	YES

Average Task Completion Time for Experiment 2

Static version	**ER-by-Design version 2.0**	**T-value**	**Reject (90% confidence)**
MEAN:66.66 STDEV:25.26 n=18	MEAN:75.26 STDEV:31.48 n=19	-0.91	NO (18%)

Average Task Completion Time for Experiment 3

Static version	**ER-by-Design version 2.0**	**T-value**	**Reject (90% confidence)**
MEAN:91.95 STDEV:95.10 n=18	MEAN:68.19 STDEV:31.48 n=19	0.99	NO (84%)

Relative Improvement in Task Completion Time (Continued)

Average Task Completion Time Comparing Experiment 1 to Experiment 3

Experiment 1		**Experiment 3**	
Static version	**ER-by-Design version 2.0**	**Static version**	**ER-by-Design version 2.0**
136.43	114.65	91.95	68.19

Relative Improvement in Task Completion Time

Static version	**ER-by-Design version 2.0**	**T-value**	**Reject (90% confidence)**
MEAN:0.24 STDEV:0.90 n=18	MEAN:0.33 STDEV:0.38 n=19	-0.38	NO (65%)

Relative Improvement in Task Completion Time for Graduates

Static version	**ER-by-Design version 2.0**	**T-value**	**Reject (90% confidence)**
MEAN:0.11 STDEV:1.39 n=7	MEAN:0.36 STDEV:0.20 n=8	-0.89	NO (80%)

Relative Improvement in Task Completion Time for Undergraduates

Static version	**ER-by-Design version 2.0**	**T-value**	**Reject (90% confidence)**
MEAN:0.47 STDEV:0.29 n=11	MEAN:0.31 STDEV:0.49 n=11	0.93	NO (18%)

Relative Improvement in Task Completion Time (Continued)

Relative Improvement in Task Completion Time for Beginning Level Group

Static version	ER-by-Design version 2.0	T-value	Reject (90% confidence)
MEAN:0.35 STDEV:0.35 n=6	MEAN:0.22 STDEV:0.66 n=6	0.42	NO (34%)

Relative Improvement in Task Completion Time for Intermediate Level Group

Static version	ER-by-Design version 2.0	T-value	Reject (90% confidence)
MEAN:0.09 STDEV:1.26 n=9	MEAN:0.39 STDEV:0.21 n=8	-0.7	NO (86%)

Relative Improvement in Task Completion Time for Advanced Level Group

Static version	ER-by-Design version 2.0	T-value	Reject (90% confidence)
MEAN:0.46 STDEV:0.19 n=3	MEAN:0.35 STDEV:0.11 n=5	0.87	NO (22%)

Relative Improvement in Amount of Help Usage

Average Number of Times Help Requested

Static version	**ER-by-Design version 2.0**	**T-value**	**Reject (90% confidence)**
MEAN:1.55 STDEV:2 n=18	MEAN:1.10 STDEV:1.59 n=19	0.75	NO (88%)

Average Number of Times Help Requested for Experiment 1

Static version	**ER-by-Design version 2.0**	**T-value**	**Reject (90% confidence)**
MEAN:1.75 STDEV:2.05 n=18	MEAN:2.9 STDEV:4.47 n=19	-1	NO (16%)

Average Number of Times Help Requested for Experiment 2

Static version	**ER-by-Design version 2.0**	**T-value**	**Reject (90% confidence)**
MEAN:1.47 STDEV:3.79 n=18	MEAN:0.06 STDEV:0.14 n=19	1.57	YES

Average Number of Times Help Requested for Experiment 3

Static version	**ER-by-Design version 2.0**	**T-value**	**Reject (90% confidence)**
MEAN:1.44 STDEV:2.18 n=18	MEAN:0.13 STDEV:0.32 n=19	2.52	YES

Relative Improvement in Amount of Help Usage (Continued)

Average Number of Times Help Requested Comparing Experiment 1 and Experiment 3

Experiment 1		**Experiment 3**	
Static version	**ER-by-Design version 2.0**	**Static version**	**ER-by-Design version 2.0**
1.75	2.97	1.44	0.13

Relative Improvement in Amount of Help Usage

Static version	**ER-by-Design version 2.0**	**T-value**	**Reject (90% confidence)**
MEAN:0.07 STDEV:0.96 n=18	MEAN:0.50 STDEV:0.51 n=19	-1.70	YES

Average Number of Times Help Requested for Males and Females

Female	**Male**	**T-value**	**Reject (90% confidence)**
MEAN:1.51 STDEV:2.62 n=10	MEAN:1.13 STDEV:1.39 n=27	-0.43	NO (77%)

Relative Improvement in Amount of Help Usage for Females

Static version	**ER-by-Design version 2.0**	**T-value**	**Reject (90% confidence)**
MEAN:-0.22 STDEV:2.1 n=3	MEAN:0.04 STDEV:0.56 n=7	-0.50	NO (33%)

Relative Improvement in Amount of Help Usage (Continued)

Relative Improvement in Amount of Help Usage for Males

Static version	**ER-by-Design version 2.0**	**T-value**	**Reject (90% confidence)**
MEAN:0.13 STDEV:0.67 n=15	MEAN:0.57 STDEV:0.50 n=12	-1.93	YES

Complexity of Final ER Diagram

Complexity of Final ER Diagram

Static version	ER-by-Design version 2.0	T-value	Reject (90% confidence)
MEAN:14.34 STDEV:4.18 n=18	MEAN:15.32 STDEV:3.85 n=19	-0.72	NO (76%)

Complexity of Final ER Diagram for Graduates

Static version	ER-by-Design version 2.0	T-value	Reject (90% confidence)
MEAN:12.71 STDEV:6.27 n=7	MEAN:15.56 STDEV:3.40 n=8	-1.07	NO (84%)

Complexity of Final ER Diagram for Undergraduates

Static version	ER-by-Design version 2.0	T-value	Reject (90% confidence)
MEAN:15.52 STDEV:1.67 n=11	MEAN: 15.15 STDEV:4.30 n=11	0.26	NO (39%)

Complexity of Final ER Diagram for Beginning Level Group

Static version	ER-by-Design version 2.0	T-value	Reject (90% confidence)
MEAN:15.08 STDEV:2.24 n=6	MEAN:14.7 STDEV:4.95 n=6	0.15	NO (43%)

Complexity of Final ER Diagram (Continued)

Complexity of Final ER Diagram for Intermediate Level Group

Static version	ER-by-Design version 2.0	T-value	Reject (90% confidence)
MEAN:14.26 STDEV:5.61 n=9	MEAN:16.15 STDEV:3.82 n=8	-0.81	NO (79%)

Complexity of Final ER Diagram for Advanced Level Group

Static version	ER-by-Design version 2.0	T-value	Reject (90% confidence)
MEAN:13.63 STDEV:2.82 n=3	MEAN:15.6 STDEV:2.98 n=5	-0.93	NO (80%)

Correctness of Final ER Diagram

Correctness of Final ER Diagram

Static version	**ER-by-Design version 2.0**	**T-value**	**Reject (90% confidence)**
MEAN:73.61 STDEV:14.12 n=18	MEAN:76.57 STDEV:12.25 n=19	-0.68	NO (75%)

Correctness of Final ER Diagram for Graduates

Static version	**ER-by-Design version 2.0**	**T-value**	**Reject (90% confidence)**
MEAN:72.14 STDEV:17.28 n=7	MEAN:74.37 STDEV:10.83 n=8	-0.29	NO (62%)

Correctness of Final ER Diagram for Undergraduates

Static version	**ER-by-Design version 2.0**	**T-value**	**Reject (90% confidence)**
MEAN:74.54 STDEV:12.52 n=11	MEAN:78.18 STDEV:13.46 n=11	-0.65	NO (84%)

Correctness of Final ER Diagram for Beginning Level Group

Static version	**ER-by-Design version 2.0**	**T-value**	**Reject (90% confidence)**
MEAN:72.5 STDEV:14.74 n=6	MEAN:85 STDEV:8.36 n=6	-1.80	YES

Correctness of Final ER Diagram (Continued)

Correctness of Final ER Diagram for Intermediate Level Group

Static version	**ER-by-Design version 2.0**	**T-value**	**Reject (90% confidence)**
MEAN:72.77 STDEV:15.83 n=9	MEAN:71.25 STDEV:12.46 n=8	0.22	NO (41%)

Correctness of Final ER Diagram for Advanced Level Group

Static version	**ER-by-Design version 2.0**	**T-value**	**Reject (90% confidence)**
MEAN:78.33 STDEV:10.40 n=3	MEAN:75 STDEV:12.24 n=5	0.40	NO (35%)

User Perception

Average User Perception Score

Static version	ER-by-Design version 2.0	T-value	Reject (90% confidence)
MEAN:5.72 STDEV:0.94 n=18	MEAN:5.70 STDEV:0.63 n=19	.067	NO (47%)

Average User Perception Score for Experiment 1

Static version	ER-by-Design version 2.0	T-value	Reject (90% confidence)
MEAN:6.11 STDEV:0.63 n=18	MEAN:5.8 STDEV:0.76 n=19	1.35	NO (9%)

Average User Perception Score for Experiment 2

Static version	ER-by-Design version 2.0	T-value	Reject (90% confidence)
MEAN:5.44 STDEV:1.33 n=18	MEAN:5.28 STDEV:1.24 n=19	0.38	NO (35%)

Average User Perception Score for Experiment 3

Static version	ER-by-Design version 2.0	T-value	Reject (90% confidence)
MEAN:5.59 STDEV:1.11 n=18	MEAN:6.02 STDEV:0.68 n=19	-1.39	YES

User Perception (continued)

User Perception Comparing Experiment 1 to Experiment 3
Based on Question

Question	**Experiment 1**		**Experiment 3**	
	Static version	**ER-by-Design version 2.0**	**Static version**	**ER-by-Design version 2.0**
1	6	5.736842105	5.611111111	6.052631579
2	6.5	5.789473684	5.888888889	6.157894737
3	6.222222222	5.894736842	5.277777778	5.684210526
4	6.055555556	5.736842105	5.611111111	6.105263158
5	5.722222222	5.631578947	5.666666667	6.105263158
6	6.111111111	5.631578947	5.444444444	6.052631579
7	6.222222222	5.894736842	5.666666667	6
Average	6.119047619	5.759398496	5.595238095	6.022556391

Improvement in User Perception

Static version	**ER-by-Design version 2.0**	**T-value**	**Reject (90% confidence)**
MEAN:-0.52 STDEV:0.89 n=18	MEAN:0.21 STDEV:0.71 n=19	-2.89	YES

User Perception for Each Question
on a Scale of 1-Least Positive to 7-Most Positive
for Session 3

Question 1: This system was very easy to use.

Static version	**ER-by-Design version 2.0**	**T-value**	**Reject (95% confidence)**
MEAN:5.61 STDEV:1.24 n=18	MEAN:6.05 STDEV:0.91 n=19	-1.22	NO (89%)

Question 2: I was successful in completing my task

Static version	**ER-by-Design version 2.0**	**T-value**	**Reject (95% confidence)**
MEAN:5.88 STDEV:1.13	MEAN:6.15 STDEV:0.68	-0.86	NO (80%)

Question 3: This system was very helpful for this task

Static version	**ER-by-Design version 2.0**	**T-value**	**Reject (95% confidence)**
MEAN:5.27 STDEV:1.36	MEAN:5.68 STDEV:1.05	-1	NO (84%)

Question 4: I was able to complete my task with a minimum level of frustration.

Static version	**ER-by-Design version 2.0**	**T-value**	**Reject (95% confidence)**
MEAN:5.61 STDEV:1.37	MEAN:6.10 STDEV:0.73	-1.34	YES

User Perception for Each Question
on a Scale of 1-Least Positive to 7-Most Positive
for Session 3 (Continued)

Question 5: I was able to complete my task quickly

Static version	ER-by-Design version 2.0	T-value	Reject (95% confidence)
MEAN:5.66 STDEV:1.13	MEAN:6.10 STDEV:0.80	-1.34	YES

Question 6: I was comfortable with the interface of the program.

Static version	ER-by-Design version 2.0	T-value	Reject (95% confidence)
MEAN:5.44 STDEV:1.38	MEAN:6.05 STDEV:0.91	-1.57	YES

Question 7: The system's level of assistance was appropriate for my level of knowledge of creating ER diagrams.

Static version	ER-by-Design version 2.0	T-value	Reject (95% confidence)
MEAN:5.6 STDEV:1.13	MEAN:6 STDEV:0.81	-1.01	NO (84%)

Improvement in User Perceived Proficiency

Improvement in User Perceived Proficiency (Question 8)

Static version	**ER-by-Design version 2.0**	**T-value**	**Reject (90% confidence)**
MEAN:0.5 STDEV:1.09 n=18	MEAN:1 STDEV:1.05 n=19	-1.41	YES

VITA

Lisa Hunt received a Bachelor of Arts degree in Psychology from Tulane University and expects to receive a Master of Science in Computer and Information Sciences from the University of North Florida, April, 1999. Dr. Krissten N. Cooper of the University of North Florida is serving as Lisa's thesis advisor. Lisa has served as an adjunct instructor in the Department of Computer and Information Sciences at the University of North Florida.

Before continuing her studies at the University of North Florida Lisa taught English through Berlitz language institute as well as pursued theatre in Los Angeles, California. She grew up in Columbia, South Carolina and over the last ten years has studied, lived and worked in New Orleans, Louisiana, San Francisco, California, Los Angeles, California, Nashville, Tennessee and, of course, Jacksonville, Florida. Lisa's academic work has included use of C, C++, SQL, PROLOG, Java and Visual Basic.