Management of Computing

Robert Zmud
Editor

Comparing Representations with Relational and EER Models

The diffusion of technology to end users who can now develop their own information systems raises issues concerning the cost, quality, efficiency, and accuracy of such systems.

Dinesh Batra, Jeffrey A. Hoffer, and Robert P. Bostrom

End-user development of information systems represents a major departure from the development of systems by trained and experienced specialists [2, 28, 34]. This trend raises numerous questions concerning the efficacy and hidden cost of such systems which may be poorly designed because of the users' lack of expertise. Major tools used by users to develop their systems are database management systems (DBMSs) and fourth generation languages with DBMS capabilities. Clearly the most popular DBMS for user application development is any of a variety of systems based on the relational data model [3]. The relational model is characterized by desirable properties like data independence and ability to support high-level query languages [26].

Another trend likely to affect the use of database management systems is the proliferation of data models. The more recent data models are generally termed *semantic data models* [31]. The diffusion of information technology to end users coupled with the propositions of semantic data models raises the human factor issue of usability of these models. This article describes such a usability study which compares representations developed by end users using two popular data models—the relational, which is a classical data model, and the extended entity relationship (EER) model, which is a semantic data model.

DATA MODELS

In the general database design literature and in the practice of professional data analysts, the relational data model [8] has been used to conceptualize data requirements [10, 26, 40]. However, several researchers have noted a shortcoming of the relational model that it is difficult to capture certain semantics of the real world. Schmid and Swenson [35] note that the relational theory gives no indication of the way in which the world is to be represented by a collection of relations. Kent [24] has catalogued and explicated the limitations of record-based information models, including the relational model. The limitations of the classical models have led to proposals for semantic data models (e.g., [4, 6, 17, 19, 31, 37, 38]) that are capable of coping with more intricate semantics inherent in many situations. It seems that the most widely accepted semantic data model is Chen's entity-relationship (E-R) model which explicitly adopts the view that the real world consists of entities and relationships [6]. Chen introduced an associated graphical representation technique as a tool for database design. Recently, the E-R model has been extended to include the notion of categories [15]. This model is called the extended entity-relationship (EER) model. Teorey et al. [41] present the EER model as a logical design tool that can be used to conceptualize data requirements. The EER representation can then be converted to a relational representation (or any other data model) for database implementation. Thus implicitly, these authors make the assumption that the EER model, as compared to the relational model, is the better representation for conceptual design. However, there is little empirical evidence that the EER model or any other semantic model leads to better novice end-user performance than the classical models in a conceptual data modeling task.

THE NOVICE USER AND DATABASE DESIGN

Database technology is now readily available for application to nontrivial problems by end users. The phenomenon of end-user computing has been supported by relational data management technology. End users may be autonomous, that is, users who design, develop, implement, and use application programs to support personal or a small group's information requirements for decision making [12, 13]. To build effective systems, these users not only need easy-to-use software, but a basic knowledge of techniques about system analysis and design as well. Since most information systems have a database component, it is important that these users understand the database design process and techniques.

The first step of database design is the development of the conceptual data model of the application. A conceptual data model is an abstraction of the real world (organizational) data pertinent to an enterprise. The process of deriving and analyzing the data inherent in a business situation and of mapping objects of this understanding of reality in a conceptual model representation constitutes the discovery phase. The discovery phase consists of two parts. The first part involves the elicitation of the information requirements. This is traditionally a result of an interview between the user(s) and the analyst(s). The second part, in the data modeling context, involves the (conceptual) representation of the information requirements into the form of a conceptual model. In this process, a data model provides representation primitives to aid in the development and specification of a conceptual model. It also provides a discipline that can highlight inconsistencies in one's qualitative understanding of a situation. In the case of end-user–developed applications, the discovery phase predominantly involves representation, and only minimally involves elicitation since the user-analyst communication gulf is eliminated.

This study concentrates on the second part of this phase, that is, the translation of one's understanding of a business situation into a representational form for a conceptual database. (We assert that even though the requirements are available, the important task of "fitting" these requirements into a conceptual model remains.) One may question the relevance of this phase and argue that some kind of a conceptual model is already available at the end of the elicitation process, albeit in a natural language. Yet, the transformation into a formal representation may not be trivial. An analogy may be drawn from the formulation of complex mathematical models which are essentially representations of problems first conceptualized and expressed in a natural language.

The purpose of this article is to report the design and results of a study that was conducted to test if the use of a semantic model instead of a relational model results in superior end-user performance. The semantic model used was the EER model. A pilot study, termed here as the first pilot study, was conducted in November 1987. The main purpose of this study was to identify any procedural problems. The data collected from the first pilot was not used for analysis. The data for the reported study was collected during February 1988 and September 1988.

PRIOR RESEARCH

A survey of human factors studies on databases suggests that most of the research has focused on programming tasks using database query languages (DBQLs). The interested reader may refer to a survey article by Reisner [32]. However, the focus of our study was on data representation rather than data manipulation. Therefore, we have not included the literature about database query languages, and present only the literature relevant to the scope of our study. Reisner's survey deserves mention, however, since it focuses on some important variables that seem to be consistently used in the human factor studies on query languages.

Most studies covered in Reisner's survey compare different query languages, so *query language* was obviously one of the variables. She stated that the methodology used to compare query languages is an extension of human factors, and, therefore, the studies include a measure of user performance as well as a set of tasks to capture some aspect of user performance. Although not explicitly, Reisner also considers the characteristics of subjects who participated in the studies. For example, some of the studies compare the performance of nonprogrammers with that of programmers. Other studies compare the performance of less experienced subjects with more experienced ones. Thus, at least implicitly, the variable *human* is considered in these studies. Reisner's survey of query language studies shows, therefore, that the laboratory studies on comparison of query languages implicitly consider the following variables: *data models*, *human*, *task*, and *user performance*. The general framework of these studies, therefore, is similar to the (explicit) framework used in other MIS experiments [22].

A literature survey on human factors studies on data models revealed that the studies can be classified into two categories—studies that compare one classical model with another classical model, and studies that compare the relational model with a semantic data model. The more recent studies fall in the second category. The studies were also scrutinized to infer if the framework which applied to query languages was also applicable to data models.

One of the earliest published studies was performed by Lochovsky and Tsichritzis [25], who compared the three classical models—hierarchical, network, and relational. Each model was implemented by using a different language: the IMS language DL/I [21], the DBTG COBOL DML [7], and ALPHA [9], respectively. Fifty-eight subjects were given query-writing tasks. Results show that for less experienced users, relational group scores are significantly better than the other two groups. Though the authors conclude that the relational model is superior, they point out that it is difficult to ascribe the results either to the data model or to the language since different models used different query languages.

To overcome the problem of query languages confounding the effects of data models, Brosey and Shneiderman [5] compared relational and hierarchical models using instance diagrams. Comprehension, problem solving situation, and memorization tasks were performed by undergraduate subjects. Significant effects are found for the data model, presentation order, subject background, and tasks. The hierarchical model was easier to use, but only for the beginning programmer group. The conceptual model used in the experiment is hierarchical in structure, and may have favored the hierarchical model.

Durding et al. [14] conducted three experiments to investigate how people organize data. This study does not use specific data models (and is, therefore, not included in Table I). Results show that the subjects organized most word sets based on semantic relations inherent in them. These results suggest that the ease of use of a model depends upon the inherent structure of data in an application. However, real world applications are generally a mix of various structures. This study did not, therefore, provide answers to whether any organization approach in general was better.

We note that the three studies just described had different implications. The Lochovsky and Tsichritzis study [25] found that the relational model is better, the Brosey and Shneiderman study [5] concludes that the hierarchical model is better, and the Durding et al. study [14] concludes that there is an interaction between the performance using a data structure and the underlying semantic structure of the application. It is clearly evident that no data model is better in all cases. In the Lochovsky study, the data manipulation task results in a better performance for the relational model. They also found that programmers did better than nonprogrammers, so human characteristics also affected the modeling performance. In the Brosey study, too, programmers performed better than nonprogrammers. Further, the description of the application (task) seemed to result in superior performance using the hierarchical model. Thus the divergent results found in these studies could be explained by considering the framework which shows user performance as dependent on data model, task, and human variables. While the framework is more implicit than explicit in these studies, it does suggest that user performance not only depends on data models, but also on the specific task and the characteristics of the designer or the user.

Hoffer [18] first reported the result of an investigation of individual differences in using database models. He found that subjects have individualized images of a database, and that a process flow structure is the most frequently used image. He also reported that subjects omit identification of database keys from their images and are not able to clearly specify data relationships. The subjects used a variety of data models which supports the earlier suggestion that a typical application would involve a mix of structures, and hence is unlikely to be naturally suited to a specific data model.

Even though these experiments did not provide any distinct conclusions about the relative ease of use of the relational *model*, it seems that the ease of use of relational *systems* is now widely accepted. The major factor responsible for this is probably the ability of the model to support a nonprocedural query interface. However, the ability of nonexpert users to successfully use the relational model for conceptual modeling is still under question.

Recent studies comparing data models use relational models as one and semantic data models as the other treatment. Since no popular commercial implementations are based on semantic data models, the comparison has been at the conceptual level only. It may be argued, however, that the relational model is no longer

TABLE I. Human Factor Studies in Data Modeling

Study	Human	Data Model	Task	Performance
Lochovsky and Tsichritzis [25]	Experience: Less experience and More experience	Relational Network Hierarchical	Query writing	Query correctness
Brosey and Shneiderman [5]	Experience: Beginner and Advanced	Relational Hierarchical	Comprehension Problem solving Memorization	Correctness
Hoffer [18]	Cognitive style Situation familiarity Situation specificity	Relational Network Hierarchical	Representation	Database image architecture Confidence Number of files
Juhn and Naumann [23]	Novice/Casual GPA, Computer experience, DBMS experience, Work experience treated as covariates	Logical data structure Entity relationship Data access diagram Relational	Validation (relationship finding, cardinality finding, identifier comprehension) Database search Data modeling	Correctness
Ridjanovic [33]	Novice/Casual	Logical data structure Relational	Discovery	Number of relationships Number of attributes
Shoval and Even-Chaime [36]	Casual	Normalization (Relational) Information analysis (binary relationship)	Representation	Correctness

pertinent for conceptual modeling, and hence such a comparison may not be fair. On the other hand, it may be inappropriate to make such a statement without some empirical evidence. In a study comparing how expert and novice designers develop relational representations, Batra and Davis [1] found that most experts and novices prepare the relational representation without using an intermediate model even though they are aware of other data models.

Studies comparing the relational model with semantic data models at the conceptual level have focused on elicitation and representation [33] and validation [23]. Shoval and Even-Chaime [36] compare two modeling techniques—normalization and information analysis—and focused on the representation phase.

Juhn and Naumann [23] focused on the user validation process in database design. They found that the graphical models (entity relationship and logical data structure) are more understandable than the relational and data access diagram in relationship existence-finding and cardinality-finding tasks. On the other hand, relational models did outperform graphical models with respect to identifier comprehension tasks.

Ridjanovic [33] conducted a lab experiment using MIS MBA students to investigate differences in the quality of data representations produced by nonexperts using the Logical Data Structure (LDS) and the Relational Data Model (RDM) formalisms. The subjects were asked to read a case, ask questions, and generate application data models which were then evaluated using an instrument developed by the researcher. Results indicated that, contrary to the author's hypotheses, the LDS subjects' questions were not relationship-driven, and the RDM subjects' questions were not attribute-driven. On comparing the two representations, it was found that there are significant differences in the number of relationships in favor of LDS and in the number of attributes in favor of the RDM group.

Shoval and Even-Chaime [36] compared two different methods for designing a database schema, normalization and information analysis (IA). The normalization method is based on the relational data model. The information analysis method is based on the binary relationship model developed by Nijssen [29]. The study involved 26 analysts who were trained to use the two methods with the structured analysis method of system analysis. There was evidence that the quality of the database schemata designed using normalization was better than that designed using IA, that normalization required less time than IA to perform, and that the analysts preferred normalization. The authors suggest, however, that the IA model may be more suitable for complex tasks.

On comparing the Juhn and Naumann [23] study with the Shoval and Even-Chaime [36] study, it seems that the results are mixed. Juhn and Naumann seem to suggest that the semantic data models are better while Shoval and Even-Chaime conclude that the relational model is better. In fact, Juhn and Naumann found that even for the same task, the results favor the graphical models on some aspects, and the relational model on another. Once again, one can conclude that it is difficult to make sweeping generalizations about the effectiveness of a data model. It is important to include the task and the human variables that interact with data model.

Our study extends the existing literature about laboratory studies comparing data models. First, it is complementary both to the Juhn and Naumann study since it considers the representation phase which is more appropriate for the end user rather than the validation phase which is more appropriate for a user-analyst scenario, and to the Shoval and Even-Chaime study since it considers the (less experienced) end user instead of the (more experienced) analyst. Finally, the study extended Ridjanovic's study by defining a detailed criterion for user performance. The study compared data models in terms of facets, which are at a level finer than the conceptual model and are discussed in the following section. The study goes beyond gross measures of user performance, like number of relationships, to a finer measure of modeling correctness. Thus, the study extends the line of research of usability issues of data modeling.

RESEARCH METHODOLOGY

Research Model

The broad purpose of this study was to compare classical and semantic models. The research model is shown in Figure 1. The independent, dependent, and control variables are now explained.

Independent Variables

Specific models had to be selected to represent each type. The relational model was selected as a classical model, and the extended entity relationship (EER), as a semantic model. The relational model was selected since it is now generally accepted that relational systems lead to significantly better user performance in query writing tasks than other classical systems [25]. Further, the relational model has been the basis for several PC-based DBMSs and other end-user development tools. The EER model [15] was selected since it is an extension of the ER model which has been a widely accepted semantic data model. Further, the EER model has been formally presented as a conceptual modeling tool [41].

Control Variables

Task and *human* were selected as the control variables in the experiment. Since the purpose of the study was to compare user performance between the relational and the EER models in the conceptual representation phase of database design, a task that required users to read a case and represent the characteristics of data as a conceptual model was selected. The user type was selected based on computer experience and mode of

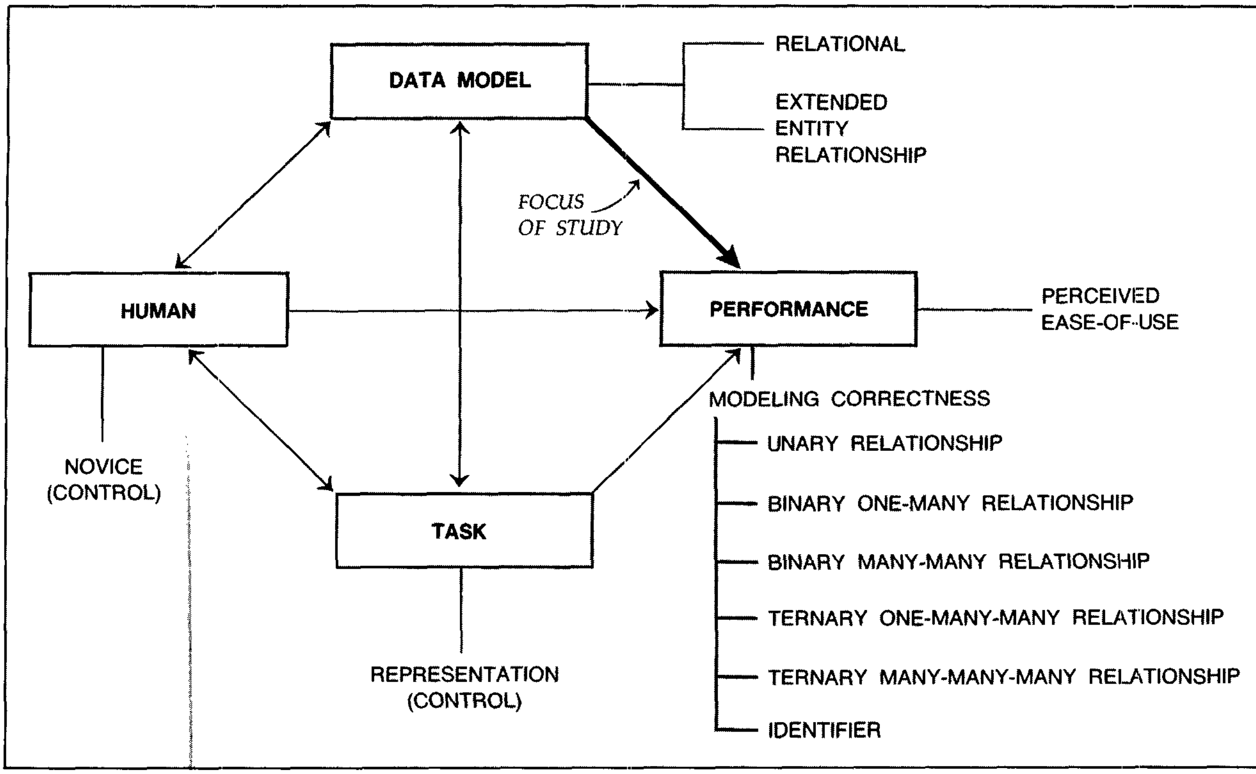


FIGURE 1. Research Model for the Study

use. This study focused on novice end users, that is, users who have limited database experience and who design their own applications. Students enrolled in an introductory MIS class were recruited and trained to use a modeling technique. The students had little or no database design background before they were trained.

Dependent Variable

The main performance variable, *modeling correctness*, can be defined as the degree to which a conceptual model approaches the correct solution(s), where the correct solution(s) convey the same semantics about data as the natural language description of the application. There was no restriction on the number of correct solutions if they translated to the same semantics. Modeling correctness was measured at the level of facets. For example, a binary one-many relationship is one facet that may occur in a conceptual data model. A short explanation of the notion of facet follows.

A data model may be considered as consisting of various constructs like entities, relationships, attributes, etc. A construct-like entity requires a fairly consistent set of modeling rules and uniform representation. Essentially, one has to be able to classify an object as an entity or an attribute. However, there is no consistent way of modeling relationships since they may differ in degree and connectivity. Representation of a relationship depends on its degree. For example, unary relationships are modeled differently than binary relationships. A change in connectivity of a relationship also changes its representation. Hence, it is not appropriate to discuss a conceptual model at the general level of relationships: one must qualify the relationships with their degree and connectivity. It is, therefore, pertinent to introduce a construct which is more detailed. This construct is termed a *facet*. Different instances of a facet have the same representation. Different facets have different representation. For example, since any instance of a many–many binary relationship is modeled the same way, a many–many binary relationship is a facet.

An important task in this study was to create a list of facets that could be included in the study. A general criterion used to qualify facets was that each facet should commonly occur in real world applications. A review of typical conceptual models found in textbooks and journal papers provided useful information. For example, relationships of degree higher than three (ternary) were not found in the literature. Another criterion was that the difficulty level of the facet should not be beyond the capability of the typical subject. The first pilot study (November 1987) was very useful in suggesting the limits of representation capabilities of the subjects. Based on these criteria, the following facets were identified as ones which could be included in the study: entity, identifier, descriptor, category, and the following types of relationships: unary, binary one–many, binary many–many, ternary one–many–many, and ternary many–many–many.

A comparison of the relational and the EER representations revealed that the facets entity, descriptor, and category were of limited interest for this study since there is no explanation to suggest that the user performance using one of the two models could be expected to be better. Therefore, this article does not include a discussion of these three facets.

The correctness scores in each of the other facets of the conceptual model (e.g., unary relationship, etc.) were graded separately. The scores on individual facets were, however, not added up to give a measure of an "overall" modeling correctness score since it would have been a function of the included facets only. There seemed to be serious construct validity concerns with forming a composite score by adding scores (arithmetically or by assigning weights) obtained in individual facets. For example, there was no justification for adding a score obtained in a binary one-many relationship with that obtained in a ternary many-many-many relationship unless the weights based on the relative complexity and the frequency of occurrence of the two facets could be specified.

Davis (1985) defines perceived ease of use as the degree to which an individual believes that using a particular system would be free of physical and mental effort. This definition, in the present context, can be modified as the degree to which an individual believes that using a data model for conceptual design would be free of mental effort. This variable was explored when the experiment was run in September 1988 by using an instrument (see inset) which was essentially adapted from a perceived ease-of-use instrument developed by Davis (1985).

Instrument Used to Determine Perceived Ease of Use

1. I found the data modeling technique cumbersome to use.
   1 2 3 4 5 6 7
   Strongly Agree Neutral Strongly Disagree
2. Using the data modeling technique was frustrating.
   1 2 3 4 5 6 7
   Strongly Agree Neutral Strongly Disagree
3. Using the data modeling technique required a lot of mental effort.
   1 2 3 4 5 6 7
   Strongly Agree Neutral Strongly Disagree
4. The data modeling technique is clear and understandable to me.
   1 2 3 4 5 6 7
   Strongly Agree Neutral Strongly Disagree
5. Overall, I found the data modeling technique easy to use.
   1 2 3 4 5 6 7
   Strongly Agree Neutral Strongly Disagree

Variables as Covariates

The study involved 42 subjects, twenty-one in each treatment. Since the subjects were randomly assigned to the treatments, it was expected that possible effects because of confounding variables will be minimal because of the averaging effect. However, one variable, database experience, was treated as a covariate since it threatened to be a serious confound if randomization was not achieved. A selection strategy was used that would maximize homogeneity in subject profile. Study subjects were solicited from various sections of an introductory course in MIS. A typical student had programming but not database design experience. While it seemed that a fair degree of randomization was achieved (Table II), the variable was, nevertheless, treated as a covariate.

TABLE II. Database Experience

Level	Rel	EER	Pooled
0	0	0	0
1	3	4	7
2	15	14	29
3	2	3	5
4	1	0	1
5	0	0	0

Notes:
0—No experience
1—Word Processing, LOTUS, etc.
2—Programming, Command Language
3—Database Design using PC DBMS
4—Formal database course
5—Expert; Formal database courses and experience

HYPOTHESES

In this section, seven hypotheses addressed by the study are presented. Five hypotheses were relationship-based, one each pertained to identifiers and perceived ease of use. The null forms of the hypotheses are not presented.

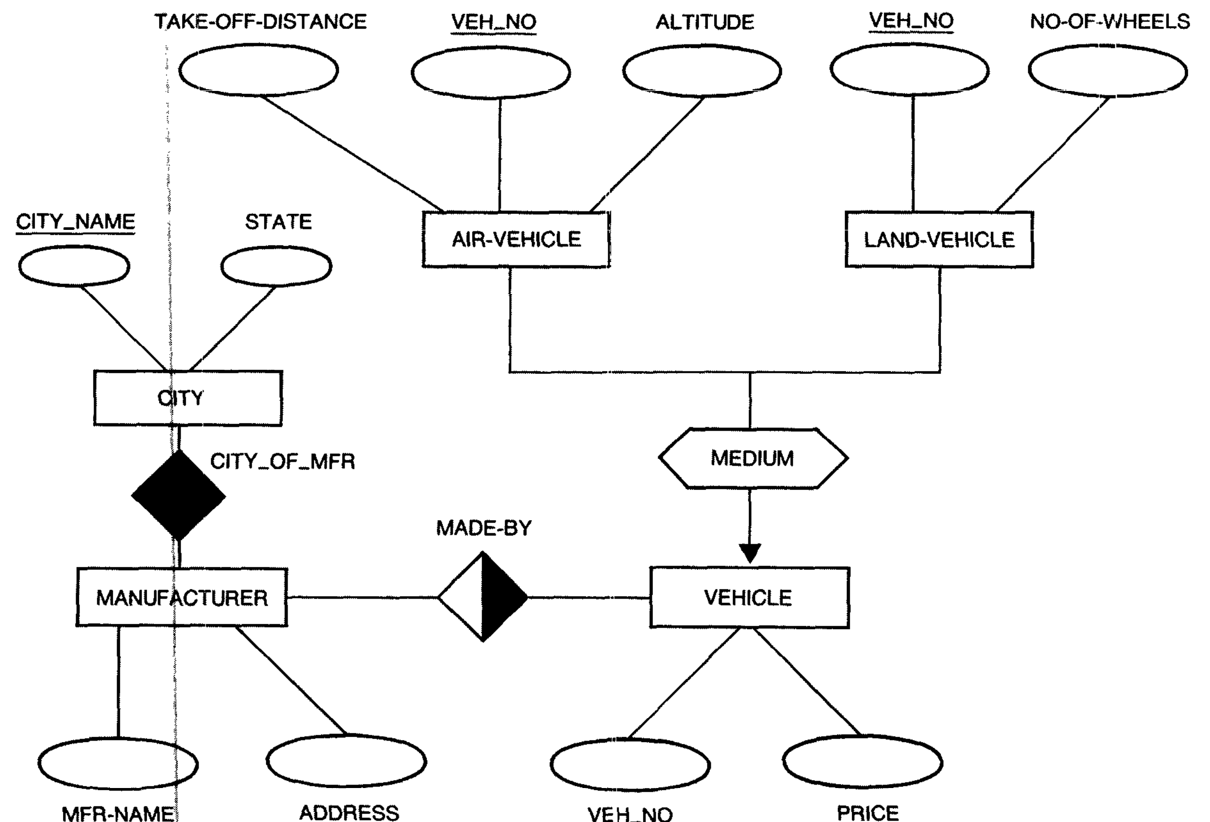
For representing a relationship and its characteristics, EER provides a direct method, that is, a notation. In contrast, the relational model accomplishes this by associating identifiers of the involved entities. This study included the following kinds of relationships: unary, binary one-many, binary many-many, ternary one-many-many, and ternary many-many-many. For all types of relationships, it was predicted that better performance would be achieved using the EER model. Thus, the hypotheses were:

The EER model, as compared to the relational model, will lead to significantly better user performance in modeling:

H1) unary relationships.
H2) binary one-many relationships.
H3) binary many-many relationships.
H4) ternary one-many-many relationships.
H5) ternary many-many-many relationships.

These relationship-based hypotheses were explained by applying the concepts developed by Hutchins, Hollan, and Norman (1985) in their model of the human-computer interface. According to their human-computer interface model, there is a gulf (or directness distance) between user's goals and knowledge of the application domain, and the level of description provided by the systems with which the person must interact. The amount of cognitive effort it takes to manipulate and evaluate a system is directly proportional to this distance. One can identify two different kinds of distances, *semantic and articulatory*, that have to be spanned between the user and the conceptual model. In the context of database design, semantic distance is concerned with the relationship of the meaning of the conceptual model to user's knowledge of real world data. Since a conceptual model is composed of objects, properties of these objects, and the association between these objects, the semantic distance relates to the distance between these real world semantics and the constructs provided by the data model for developing the conceptual data model. Articulatory distance is related to the meaning of the conceptual data model and its physical form. Thus, a relational model has a tabular form, while the EER model has a pictorial form.

It was hypothesized that the EER model would facilitate lower semantic distance than the relational model because it captures the characteristics of the relation-



EER REPRESENTATION

VEHICLE (VEH_NO, PRICE, MEDIUM, MFR_NAME)

CATEGORIES OF VEHICLE BY MEDIUM

LAND_VEHICLE(VEH_NO, NO_OF_WHEELS)
AIR_VEHICLE(VEH_NO, ALTITUDE, TAKE_OFF_DISTANCE)

MANUFACTURER(MFR_NAME, ADDRESS)
CITY(CITY_NAME,STATE)
CITY_OF_MFR(MFR_NAME, CITY-NAME)

RELATIONAL REPRESENTATION

FIGURE 2. Differences in Representations (from [38])

ships between entities in a more direct fashion. In fact, the EER model has a special symbol for a relationship which captures its degree and connectivity. The relational model, on the other hand, captures relationships in a more complicated manner, and will lead to a larger semantic distance. For example, Figure 2 shows a problem represented using the relational and the EER model. The relationships MADE_BY and CITY_OF_MFR are both binary relationships. However, the connectivity of MADE_BY is one-many, and CITY_OF_MFR is many-many. In the EER representation, the symbols representing the relationships are similar except that they reflect differences in connectivity. The relational representation shows less uniformity. The relationship MADE_BY is not shown explicitly, but is implicitly captured by embedding the identifier MFR_NAME in the relation VEHICLE (since the key VEH_NO can functionally determine the key MFR_NAME). However, the relationship CITY_OF_MFR requires that a separate relation be defined (since, between MFR_NAME and CITY_NAME, no one key can functionally determine the other). Thus there is lack of a consistent rule in modeling relationships. Using the relational model, the problem is likely to get more acute as the degree of the relationship is increased since there will be more possible combinations of connectivity, and therefore a larger number of rules to represent the relationships. This may confuse a nonexpert user.

It was also hypothesized that the EER model, as compared to the relational model, would lead to lower articulatory distance. An EER representation has a graphical form (articulation) as opposed to the relational representation which has a textual/tabular, unidimensional form. Of course, as has been frequently documented in the MIS literature, there is no guarantee that a graphical form, as compared to a tabular form, would lead to superior performance. The superiority of a form is essentially a function of task. In this particular context, it was hypothesized that the availability of a graphical formalism would facilitate modeling of relationships. This argument is based on two main reasons. First, in an EER representation, a relationship is always shown explicitly between the objects. Thus, if two real world objects, like VEHICLE and MANUFACTURER, have a relationship between them, then the representation for the relationship is shown connecting the representations for the entities involved in the relationship. However, in the relational representation, the relationship is represented by associating the identifiers of the objects, and not representations of the objects themselves. Second, since a relationship, by its very definition, is an association between objects, the connection of objects by graphically linking them in an EER representation is a more direct way of showing the relationship. There are no explicit links in a relational representation.

The hypothesis on modeling of identifiers was formulated in favor of the relational model. In either of the models, an identifier serves to uniquely distinguish instances of an entity. However, in the relational model, identifiers are also used to define the relationships between entities. Therefore, it was expected that there would be better discipline in specifying identifiers using the relational model.

This difference may be explained by the notion of a forcing function [30]. A forcing function is present when there is some feedback from the world that prevents an operation from taking place if it were being done erroneously. Thus, the design of a system should "force" a certain sequence of actions so that errors are prevented. In the relational model, relationships are defined using identifiers of the involved entities, therefore, the representation of the relationships forces the definition of the identifiers.

> H6) The relational model, as compared to the EER model, will lead to significantly better user performance in specifying identifiers of the respective entities

It was hypothesized that the same reasons which lead to better user performance in modeling relationships would also lead to the perception that the EER model is easier to use. The relationships seem to be the most difficult component in conceptual modeling since they capture a lot of semantics. The EER model, with a more direct approach of modeling relationships, was expected to be, therefore, easier to use.

> H7) Users would perceive EER model, as compared to the relational model, as higher in ease of use.

Research Strategy

A laboratory study was conducted to address the research question. Traditionally, the laboratory has been the setting for most human factor research. In fact, the laboratory-based research strategy has been used in all data modeling studies comparing user performance. The main advantages of the laboratory method are high internal validity [39], precise definition and manipulation of independent variables over feasible ranges [16], and control for nuisance variables or extraneous variables [27].

The task required subjects to develop a representation using one of the two models. A description of the application to be represented was adapted from the example in [41] and is presented in Appendix A. The relational and the EER solutions are presented in Appendix B and Figure 3, respectively. Recall that any solution semantically equivalent to these was also acceptable. The application description was not biased in favor of any of the models since it includes most of the facets typically found in database design, that is, the task is not restricted to one or few facets. The description of the problem was presented in text form. It was devoid of any pictures or tables. Although the problem is nontrivial, the domain of the problem (which deals with an employee database) is simple. These precautions were intended to avoid bias in favor of any model. The problem was tested in the first pilot for any ambi-

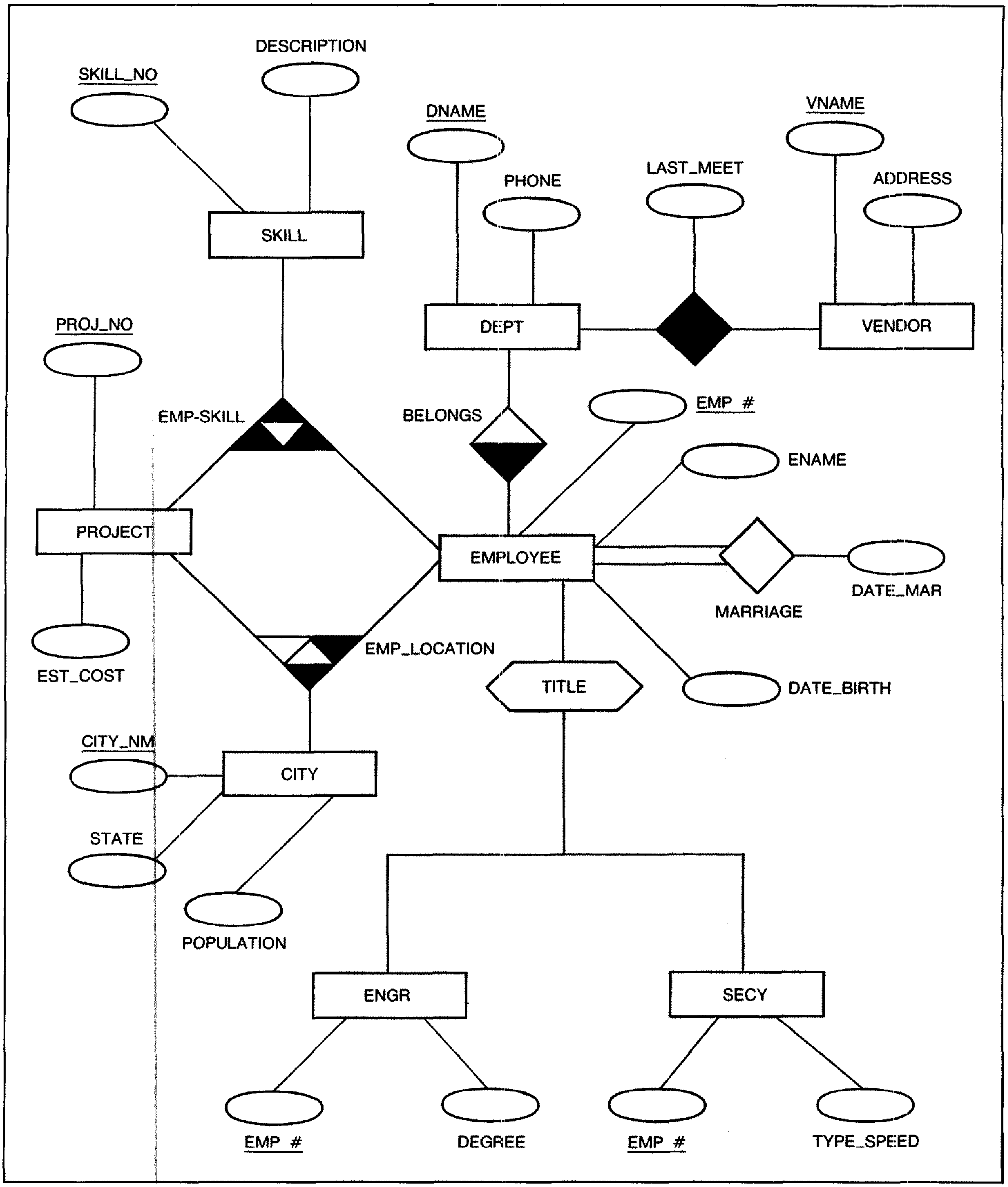


FIGURE 3. EER Solution

guities and lack of clarity, and based on the feedback from the subjects, was enhanced and used in the main study.

Graduate students were recruited mainly from introductory MIS courses. Forty-two subjects participated in the study. The subjects came from a variety of majors. There was no monetary incentive to participate in the study. The incentive offered to subjects was the training in certain aspects of database design. Participation in the experiment was voluntary, and subjects could withdraw from the experiment at any time. The investigator observed fairly high motivation.

The database experience of the subjects (from either prior work or education) was controlled to a fair extent by the selection process. Since the subjects were recruited from introductory courses in MIS, a large majority of them had had no formal database courses. The subjects filled in a questionnaire that collected data about their computer experience. Then based on their database experience they were placed into one of the six categories (Table II). Subjects using the relational model had a mean score of 2.0, while those using the EER had a mean score of 1.96. This suggests that randomization averaged out any individual differences in database design. Further, since database experience is an important variable, it was also treated as a covariate. The variations in database experience were measured before any training. Explicit training of subjects on the conceptual representation task further minimized differences in database experience. An analysis of covariance using database experience as a covariate revealed that the variable did not seem to significantly affect user performance.

A few days before the laboratory session, consenting subjects were provided with a short note "Conceptual Modeling," which introduced them to the basic terminology generally used in database design. Each subject was given an appointment for the experiment. Appointments were allotted so that typically four subjects would participate in an experimental session.

The actual experiment had the following sequence:

1. The subjects were asked to complete a questionnaire related to personal demographics and computer experience.
2. They were then provided with a set of notes and trained by the experimenter for approximately 45 to 50 minutes in using one of the models. The training notes for each model had been reviewed by database faculty, to ensure that the script was not biased in favor of any model. Further, two independent reviewers sat through some of the training sessions, and rated the equivalence of the training. This was done to check for any experimenter bias.
3. The subjects were asked to develop, in a suggested time of 30 minutes, a conceptual model for an employee database. They were not forced to stop after the suggested time had elapsed. They were provided with a textual description of the problem and allowed the use of the training notes to complete the task.
4. After each subject had finished the task, a debriefing questionnaire was provided to the subject so that he/she could provide feedback and report any ambiguities in the exercise.

Grading Scheme

The representation prepared by each subject was graded for correctness by three raters by comparing it with the correct solution. The problem and the solution were adapted from [41]. The grading scheme (Table III) was designed to provide maximum consistency of scoring between the two models, and with the data modeling training. The raters were provided with the grading scheme and supporting explanations of the scheme.

Each facet was graded separately. A score of 1 was awarded for each correct facet and 0 for incorrect or missing facet. Partial credit was given for partially correct representations. To facilitate this, errors were classified as *minor*, *medium*, or *major*, and 0.25, 0.50, and 0.75 were deducted, respectively.

The grading scheme was developed primarily from an analysis of errors found in the solutions prepared by subjects in the first pilot. The frequently occurring errors were categorized based on the severity as *minor*, *medium*, or *major*. The scheme was evaluated by faculty experienced in teaching database courses, and any proposed changes were discussed, and if necessary, implemented. When the final scheme was formulated, it was found that none of the frequently occurring errors could be classified as major errors. Thus, the grading scheme (Table III) shows minor and medium errors only. Errors found in the representations prepared by the subjects and not covered by the grading scheme were classified as minor, medium, or major depending on the rater's perception of severity of the errors.

RESULTS AND INTERPRETATIONS

Three raters independently graded the representations prepared by subjects. The final scores so obtained were averaged over the three raters and then converted to

TABLE III. Grading Scheme for the Study

Item	Incorrect	Medium Error	Minor Error
Relationships	Missing Incorrect degree	Incorrect connectivity but correct degree Unary relationship captured by categories but categorizing attribute not shown	Unary relationship captured by using an attribute (EER model only) Employing entity names instead of identifiers (relational model only)
Identifiers	Missing Identifier different from the one specified in the task description	Attribute not underscored	

percentages. The overall inter-rater reliability (Cronbach's alpha) was found as 0.97. The results from the comparison of means of facets and perceived ease of use are shown in Table IV. The mean correctness scores are in percentages. The study used a critical significance level (α) for each test of facet means as 0.01. There was little difference between the time taken by the relational (36.1 minutes) and the EER (35.3 minutes) groups to complete the exercise.

The unary (hypothesis H1)—that the EER model, as compared to the relational model, would lead to a significantly higher score—was not supported. In fact, the relational group scored higher by 13.1 percent, although this did not result in significance ($p = 0.26$). It was felt that the higher score in case of the relational model may be because of a more concrete method of modeling unary relationships. For example, in the case given to the subjects, a unary relationship was required to capture the following semantics: "If an employee is married to another employee of Projects Inc., then it is required to store the date of marriage and who is married to whom. However, no record need be maintained if the spouse of an employee is not an employee of the firm." Even though the relationship involves only one entity, an instance of a relationship involves distinct instances (the employee and the spouse who is also an employee) of the EMPLOYEE entity. In a relational model, this can be captured by the following representation:

MARRIAGE(<u>EMP#,</u> SPOUSE#, DATE_OF_MAR)

As can be seen above, this is concretely captured by EMP# and SPOUSE# in the relational model. However, in case of the EER model, it is captured by a relationship symbol with two links to one entity (refer to Figure 3). This hides the fact that the relationship is between two instances of the same entity. This was also evident by the fact that some subjects showed SPOUSE as a separate entity and then showed the marriage relationship as binary.

Both hypotheses H2 and H3, one-many and many-many (binary) relationships were supported. Subjects using the EER model made considerably fewer errors. The significance level for the binary one-many hypothesis was 0.0035, while that for the binary many-many was 0.0007.

The results clearly note the inadequacy of the relational model for capturing binary relationships. In fact, the problems about the way the relational model captures relationships were evident. First, there is an inconsistency problem since the connectivity of the relationship dictates if it will be captured explicitly (e.g., many-many), or implicitly (e.g., one-many). A number of connectivity errors were found. Second, the relationships are represented by associating the identifiers of the involved entities, not the entities themselves. It was found that subjects occasionally attempted to capture a relationship by using the entity names, not their identifiers. Another error that was found was the reciprocal inclusion of identifiers, that is, the relationship was captured between two entities by including the identifier of each entity in the relation for the other entity.

The inadequacy of the subjects using the relational model to represent binary relationships is a significant finding since these seem to be the most common types of relationships found in most databases. Similar errors were rarely found in the EER representations prepared by the subjects.

The score was significantly higher for the EER group in case of the ternary one-many-many relationship ($p = 0.0004$), but not for the ternary many-many-many relationship ($p = 0.33$). The most important observation was the sharp fall in the scores in general as compared to scores pertaining to binary relationships.

These results suggest three important points. First, for novice end users, ternary relationships are difficult to model. In fact, one should not expect such users to model relationships of degree higher than 3. Second, relationships where the connectivity is partly one and partly many seem to be more difficult to model. Finally, the EER model seems to lead to better user performance although this was not fully supported since the hypothesis H5 pertaining to the many-many-many relationship was not statistically significant.

Regarding the identifiers (hypothesis H6), the mean score for the relational model was 72.4 and that for the EER model was 73.9. There was no significant difference between the mean scores of the two treatment

TABLE IV. Results of the Study

Hypothesis[a]	Facet	Mean Relational[b]	Mean EER[b]	Significance Level	Hypothesis Support
H1	Unary Rel	68.3	55.2	0.26	No
H2	Binary One-Many Rel	54.4	84.9	0.0035[c]	Yes
H3	Binary Many-Many Rel	57.1	92.9	0.0007[c]	Yes
H4	Ternary One-Many-Many Rel	8.33	41.3	0.0004[c]	Yes
H5	Ternary Many-Many-Many Rel	33.3	45.2	0.33	No
H6	Identifiers	72.4	73.9	0.82	No
H7	Perceived Ease-of-Use	3.78	3.42	0.50	No

Notes:
[a] Hypotheses H1 thru H6 are based on a sample size of 42; hypothesis H7 is based on a sample size of 19.
[b] Mean scores are in percentages
[c] Significantly different scores

groups. This was counter to the hypothesis which predicted higher score for the relational group. In fact, both groups seemed to perform well. It may seem somewhat contrary to Hoffer's finding which reported that subjects frequently omitted specifications of identifiers. It is felt that the training imparted to the subjects, which stressed specifications of identifiers, was probably responsible for our results. Another reason for the good performance may be the application description, which in most cases specified or suggested an identifier. It may be interesting to see the results if no identifier is indicated in the application description.

Perceived ease of use of a data model is the degree to which an individual believes that using the data model would be free of mental effort. The hypothesis that the EER model, as compared to the relational model, would lead to greater perceived ease of use (hypothesis H7) was not supported. On a scale that represented highest ease of use at 1, and lowest ease of use at 7, the mean score of the EER group was 3.42, and the mean score of the relational group was 3.78. A t-test indicated that this difference had a significance level of 0.50, which was not significant. This shows that, for the category of users considered in the study, the relational model was perceived as more difficult to use. However, the difference was not large enough to be of statistical significance. The reliability of the instrument used to measure perceived ease of use was 0.83.

Overall Interpretation of the Results

The overall objective of the study was to identify the better data model for end users engaged in conceptual representation task. The EER model scored higher in the correctness score of all facets except for unary relationship. Given that the unary relationship is rarely encountered anyway, the overwhelming evidence is in favor of the EER model. The most notable error found in the solutions prepared by the subjects was the incorrect representation of connectivity of relationships. There was no difference found in perceived ease of use of the two models. Thus, the EER model was found to lead to better performance, but was not perceived as significantly easier to use than the relational model. One reason why a statistically significant result for perceived ease of use was not obtained was the low sample size (19 subjects) for this test as compared to the other tests of the experiment (42 subjects). Further, because of the between-subjects design of the experiment, the subjects could not compare their perceptions of ease of use of the two data models.

IMPLICATIONS AND FURTHER RESEARCH

The general evidence from the study was that user performance in a representation task using the EER model, as compared to the relational model, was better. The EER model led to significantly better user performance in modeling binary one–many, binary many–many and ternary one–many–many relationships. As the degree of the relationship increased from binary to ternary, there was a sharp decline in user performance. The sharp deterioration of user performance in modeling ternary relationships probably sets limits to the degree of relationships that can be successfully modeled by nonexpert end users.

Our study provides some clue about whether novice end users can develop conceptual models if they understand the application. The performance of the subjects in most facets, especially using the EER model, was satisfactory enough to suggest that this is a real possibility. Some of the errors found in the study would usually be caught as the database is defined via a DBMS. A follow-up field study is needed to better understand what types of errors or inadequacies are actually implemented by novice end users.

The results from the study also have practical significance. Currently, end users are trained only to use DBMS software packages which are generally based on the relational model. However, for effective use of such software, there is a need to train and support users in the discovery and validation tasks. Our research along with other findings [23] suggest that semantic models, e.g., the EER model, provide better mechanisms to support many of these tasks.

These findings have many implications. The developers of DBMS can build the interface which is EER based, so that users can directly implement the EER representation, without a conversion to the relational representation. Alternatively, the novice end users can be trained to develop a conceptual model using the EER model which can then be converted automatically into a relational representation by utility software. Further, the developers of DBMS can provide tutoring systems which assist the conceptual modeling process. Intelligent DBMS can be developed which can detect commonly occurring errors, and suggest corrections.

This study suggests that the most commonly occurring errors pertain to the connectivity of relationships. This is especially the case with ternary relationships. Thus, the training of novice end users in the representation phase should focus on such frequently occurring errors, especially if such users may be expected to design nontrivial applications (that is, applications involving several entities and relationships between these entities).

We suggest several extensions to our research. An obvious extension is to validate the research in a field setting. One can obtain actual implementations of databases designed and implemented by end users, and compare the errors found in these implementations with those found in this study. Another extension of this research would be to compare expert and nonexpert designers, so that one could identify the nature of the expertise in this context. This would also provide useful information for building tutoring and expert systems to assist the conceptual modeling phase. Finally, we need to consider the effect of other data models, other tasks, and other human characteristics on modeling performance.

APPENDIX A. EXERCISE

Projects Inc. is an engineering firm with approximately 500 employees. A database is required to keep track of all employees, their skills, projects assigned, and departments worked in. Every employee has a unique number assigned by the firm. It is required to store his/her name and date-of-birth. If an employee is currently married to another employee of Projects Inc., then it is required to store the date of marriage and who is married to whom. However, no record of marriage need be maintained if the spouse of an employee is not an employee of the firm. Each employee is given a job title (e.g., engineer, secretary, foreman, etc.). We are interested in collecting more data which is specific to the following types: engineer and secretary. The relevant data to be recorded for engineers is the type of degree (e.g., electrical, mechanical, civil, etc.) and for secretaries is their typing speeds. An employee does only one type of job at any given time and we need to retain information material for only the current job for an employee.

There are 11 different departments, and each has a unique name. An employee can report to only one department. Each department has a phone number.

To procure various kinds of equipment, each department deals with many vendors. A vendor typically supplies equipment to many departments. It is required to store the name and address of each vendor, and the date of last meeting between a department and a vendor.

Many employees can work on a project. An employee can work in many projects (e.g., Southwest Refinery, California Petrochemicals, etc.), but can be assigned to only one project in a given city. For each city, we are interested in its state and population. An employee can have many skills (e.g., preparing material requisitions, checking drawings, etc.), but he/she may use only a given set of skills on a particular project. (For example, an employee MURPHY may prepare requisitions for Southwest Refinery project, and prepare requisitions as well as check drawings for California Petrochemicals.) An employee uses each skill that he/she possesses in at least one project. Each skill is assigned a number. A short description is required to be stored for each skill. Projects are distinguished by project numbers. It is required to store the estimated cost of each project.

APPENDIX B. SOLUTION USING RELATIONAL MODEL

EMPLOYEE(<u>EMP#,</u> ENAME, DATE_BIRTH, SPOUSE#, DATE_MAR, DNAME, TITLE)
Categories of EMPLOYEE by TITLE:
EMP_ENGR(<u>EMP#,</u> DEGREE)
EMP_SECY(<u>EMP#,</u> TYPE_SPEED)
DEPT(<u>DNAME,</u> PHONE)
VENDOR(<u>VNAME,</u> ADDRESS)
DEALS(<u>DNAME, VNAME,</u> LAST_MEET)
SKILL(<u>SKILL#,</u> DESCRIPTION)
PROJECT(<u>PROJ#,</u> EST_COST)
EMP_SKILL(<u>EMP#, SKILL#, PROJ#</u>)
CITY(<u>CITY NM,</u> STATE, POP)
EMP_LOCATION(<u>EMP#, CITY-NM,</u> PROJ#)

REFERENCES

1. Batra, D., and Davis, J.G. Conceptual database design by novice and expert database designers. In *Proceedings of the Tenth International Conference on Information Systems* (Boston, Mass., Dec. 4–6 1989). p. 91–99.
2. Benjamin, R.I. Information technology in the 1990's: A long range planning scenario. *MIS Q.* (June 1982), vol. 6, 2 11–32.
3. Blaha, M.R., Premerlani, W.J., and Rumbaugh, J.E. Relational database design using an object-oriented methodology. *Commun. ACM 31*, 4 (Apr. 1988), 414–427.
4. Brodie, M.L. On the development of data models. In M.L. Brodie et al., Eds., *On Conceptual Modelling*. Springer-Verlag, New York, 1984.
5. Brosey, M., and Shneiderman, B. Two experimental comparisons of relational and hierarchical database models. *Int. J. Man-Machine Studies 10*, (1978), 625–637.
6. Chen, P.P. The Entity-Relationship model—Toward a unified view of data. *ACM Trans. Database Syst. 1*, 1 (Mar. 1976), 9–36.
7. CODASYL Data Base Task Group, April 1971 Report, ACM, New York.
8. Codd, E.F. A relational model of data for large shared data banks. *Commun. ACM 13*, (June 1970), 377–387.
9. Codd, E.F. A data base sublanguage founded on the relational calculus. In *Proceedings of the ACM SIGFIDET Workshop on Data Description, Access and Control* (Nov. 1971), pp. 35–68.
10. Date, C.J. An introduction to database systems. Vol 2, Addison-Wesley, Reading, Mass., 1983.
11. Davis, F.D. A technology acceptance model for empirically testing new end-user information systems: Theory and results. Ph.D. dissertation, Massachusetts Institute of Technology, Sloan School of Management. 1985.
12. Davis, J.G. A typology of management information systems users and its implications for effectiveness research. Ph.D. dissertation, University of Pittsburgh, Graduate School of Business 1986.
13. Davis, J.G., and Srinivasan, A. Incorporating user diversity into information systems assessment. In *Information Systems Assessment*, North-Holland. 1988, 83–100.
14. Durding, B.M., Becker, C.A., and Gould, J.D. Data organization. *Human Factors 19*, 1 (Feb. 1977), 1–14.
15. Elmasri, R., Weeldreyer, J., and Hevner, A. The category concept: An extension to the Entity-Relationship model. *Data Knowledge Eng. 1*, 11 (June 1985), 75–116.
16. Fromkin, H.L., and Streufert, S. Laboratory experimentation. In M.D. Dunnette, Ed., *Handbook of Industrial Psychology*. Rand McNally, Chicago, 1976, 415–465.
17. Hammer, M., and McLeod, D. Database description with SDM: A semantic database model. *ACM Trans. Database Syst. 6*, 3 (Sept. 1981), 351–386.
18. Hoffer, J.A. An empirical investigation into individual differences in database models. In *Proceedings of the Third International Conference on Information Systems*. (Ann Arbor, Mich., Dec. 13–16, 1982), pp. 153–168.
19. Hull, R., and King, R. Semantic database modeling. *ACM Comp. Surv. 19*, 3 (Sept. 1987), 201–260.
20. Hutchins, E.L., Hollan, J.D., and Norman, D.A. Direct Manipulation Interfaces. *Human Comp. Interact. 1* (1985), 311–338.
21. *IBM information management system/virtual storage (IMS/VS), General Information manual*. GH20-1260-3, IBM Corporation, White Plains, NY, 1975.
22. Jenkins, A.M. *MIS Decision Variables and Decision Making Performance*. UMI Research Press, Ann Arbor, Mich., 1982.
23. Juhn, S., and Naumann, J.D. The effectiveness of data representation characteristics on user validation. In *Proceedings of the Sixth International Conference on Information Systems* (Indianapolis, Ind., 1985), pp. 212–226.
24. Kent, W. Limitations of the record based information models. *ACM Trans. Database Syst. 4*, 1 (Mar. 1979), 107–131.
25. Lochovsky, F.H., and Tsichritzis, D.C. User performance considerations in DBMS selection. In *Proceedings of ACM SIGMOD* (Toronto, Canada, Aug. 3–5, 1977), pp. 128–134.
26. McFadden, F.R., and Hoffer, J.A. *Data Base Management*. Benjamin Cummings, Menlo Park, Cal., 1988.
27. McGrath, J.E. Towards a "Theory of Method" for research on organizations. In R.T. Mowday, et al., Eds., *Research in Organizations: Issues and Controversies*, Goodyear Publishing Co., Santa Monica, Cal., 1979.

28. McLean, E.R. End users as applications developers. *MIS Q.* (Dec. 1979), 37–46.
29. Nijssen, G.M. An architecture for knowledge base systems. In *Proceedings of SPOT-2 Conference.* (Stockholm, Sweden, Sept. 1981).
30. Norman, D.A. Forcing functions. Working Paper, University of California, San Diego, October 16, 1984.
31. Peckham, J., and Maryanski, F. Semantic data models. *ACM Comp. Surv. 20*, 3 (Sept. 1988), 153–189.
32. Reisner, P. Human factor studies of database query languages. *Comp. Surv. 13*, 1 (Mar. 1981), 13–31.
33. Ridjanovic, D. Comparing quality of data representations produced by nonexperts using logical data structure and relational data models. Ph.D. dissertation, University of Minnesota, Carlson School of Management 1986.
34. Rockart, J.F., and Flannery, L.S. The management of end user computing. *Commun. ACM 26*, 10 (Oct. 1983), 776–784.
35. Schmid, H.A., and Swenson, J.R. On the semantics of the relational data model. In *Proceedings of the 1975 SIGMOD Conference* (San Jose, Cal., May 1975), pp. 211–223.
36. Shoval, P., and Even-Chaime, M. Database schema design: An experimental comparison between normalization and information analysis. *Database 18*, 3 (Spring 1987), 30–39.
37. Smith, J.M., and Smith, D.C.P. Database abstractions: Aggregation. *Commun. ACM 20*, 6 (June 1977a), 405–413.
38. Smith, J.M., and Smith, D.C.P. Database abstractions: Aggregation and generalization. *ACM Trans. Database Syst. 20*, 2 (June 1977b), 105–133.
39. Stone, E. *Research Methods in Organization Behavior.* Scott, Foresman and Co., Glenview, Ill., 1978.
40. Teorey, T.J., and Fry, J.P. *Design of Database Structures.* Prentice-Hall, Englewood Cliffs, NJ, 1982.
41. Teorey, T., Yang, D., and Fry, J.P. A logical design methodology for relational databases using the extended Entity-Relationship model. *ACM Comp. Surv. 18*, 2 (June 1986), 197–222.

CR Categories and Subject Descriptors: H.2.1 [**Database Management**]: Logical Design—*data models*

General Terms: Human Factors

Additional Key Words and Phrases: Conceptual design, relational and entity relationships, representation

ABOUT THE AUTHORS:

DINESH BATRA is an assistant professor in the College of Business Administration at Florida International University. His research interests include human factors and database management. Author's Present Address: College of Business Administration, Florida International University, University Park, Miami, FL 33199.

JEFFREY A. HOFFER is a professor in the School of Business at Indiana University. His research interests include database management, strategic data and systems planning, human factors, and data security. Author's Present Address: Department of Operations and Systems Management, School of Business #BU 463, Indiana University, Bloomington, IN 47405.

ROBERT P. BOSTROM is an associate professor in the Department of Management at the University of Georgia. His research interests include support systems, end-user computing, user-designer relationships, effective design of organizations via integrating human/social and technological dimensions. Author's Present Address: Department of Management, Brooks Hall, University of Georgia, Athens, GA 30602.

Hoadley *(continued from p. 125)*

(color and black and white) on information acquisition and retrieval. *ECTJ 31*, 1 (Spring 1983), 9–21.
22. Lauer, T.W. The effects of variations in information complexity and form of presentation on performance for an information extraction task. Doctoral dissertation, Indiana Univ., July 1986.
23. Lauer, T.W., Davis, L.R., Groomer, M.S., Jenkins, A.M., and Yoo, K. *Establishment of the content validity of a metric of information set complexity.* Discussion Paper #291, Indiana Univ., July 1985.
24. Marcus, A. The ten commandments of color: a tutorial. *Computer Graphics Today* 3, 11 (Nov. 1986), 7, 12, 14.
25. Pachella, R.G., Smith, J.E.K., and Stanovich, K.E. Qualitative error analysis and speeded classification. In *Cognitive Theory: Volume 3*, (N. John Castellan, Jr. and Frank Restle, Eds. Lawrence Erlbaum Assoc., Pubs., Hillsdale, NJ, 1978, pp. 169–198.
26. Paller, A. They still need the numbers. *Information Center* 2, 11 (Nov. 1986), pp. 48–49.
27. Robertson, P.J. A guide to using color on alphanumeric displays. IBM Corp., Technical Report G320-6296-0, White Plains, NY, 1980.
28. Tufte, E. *The Visual Display of Quantitative Data*, Graphics Press, Cheshire, Conn., 1983.
29. Vogel, D., Dickson, G.W., and Lehman, J. Presentation persuasion: the impact of computer-generated visuals. Working paper, Univ. of Minnesota, June 1986.
30. Yoo, K. The effects of question difficulty and information complexity on the extraction of data from an information presentation. Ph.D. dissertation. Indiana Univ., Dec. 1985.

CR Categories and Subject Descriptors: D.2.2 [**Software Engineering**]: Tools and Techniques—*user interfaces*; H.1.2 [**Models and Principles**]: User/Machine Systems—*human factors and human information processing*; H.4.2 [**Information Systems Application**]: Types of Systems—*decision support*; I.3.7 [**Computer Graphics**]: Three-Dimensional Graphics and Realism—*color, shading, shadowing and texture*

General Terms: Computer Graphics, Color, Human Information Processing, Decision Support Systems

Additional Key Words and Phrases: Complexity, information extraction, theory development

ABOUT THE AUTHOR:

ELLEN D. HOADLEY is an assistant professor of Management Information Systems in the Joseph A. Sellinger, S.J. School of Business and Management, at Loyola College, in Baltimore. Her current research interests include the human/computer interface, specifically the areas of color and graphics. Author's Present Address: Joseph A. Sellinger, S.J. School of Business and Management, Loyola College, 4501 N. Charles St., Baltimore, MD 21210. (hoadley@loyvax).