# Using the Data Modeling Worksheet to Improve Novice Data Modeler Performance

**Douglas B. Bock**
**Susan E. Yager**
Computer Management and Information Systems Department
Southern Illinois University Edwardsville
Edwardsville, IL, USA
dbock@siue.edu syager@siue.edu

**ABSTRACT**

This research reports on use and evaluation of the data modeling worksheet as a pedagogical tool for improving a student's ability to learn the extended entity-relationship data modeling methodology. A laboratory experiment using a modified post-test only, control group design compared the performance of two student subject groups. One group used the data modeling worksheet as an integral component of their instruction on database design. A second control group did not use the worksheet, but that group received comparable training in every other respect. Subjects were tasked to develop a data model that represented a textual description of a data modeling problem. The data analysis used a one-way ANOVA to evaluate eight hypotheses, each representing a facet of the data modeling methodology. The results indicate that the data modeling worksheet significantly improved student learning with regard to their ability to identify entities, entity identifier attributes, and ternary relationships.

**Keywords**: Data model, ER Model, EER model, UML model, data modeling worksheet

## 1. INTRODUCTION

Teaching new students to model the data that supports business decision-making is a challenging task. The cognitive psychology literature presents the usefulness of an iterative, pattern matching strategy for problem solving (e.g., Gick & Holyoak, 1983; Phye, 1990; Reed, 1972; Tversky & Gati, 1989); and data modelers often match patterns in an iterative fashion (Reed, 1972). To accomplish this, however, students must learn to extract information from a modeling problem in order to develop an abstract representation of business data. At the same time, they are also learning the nuances of a particular data modeling methodology, such as the Entity-Relationship (ER) or Unified Modeling Language (UML) model. Their ability to deal with abstract concepts is in part a function of their prior learned experience in solving problems. In order to solve problems, individuals must first understand the problem by (1) identifying pertinent information and (2) then being able to "represent the problem features by an internal representation that truly describes the problem" (Konar, 2000). A good representation or abstraction of a problem can serve as an efficient solution to that problem (Konar, 2000). In this research we focus on improving student performance in dealing with abstract concepts by proposing the use of a simple organizing tool, the data modeling worksheet, during instruction on data modeling.

We first examine the nature of the data modeling task in part 2. Part 3 details the theoretical basis for use of the data modeling worksheet while part 4 details how to use the worksheet. Part 5 covers the research methodology including the research design, experimental variables, hypotheses, experimental procedure, and performance evaluation measures. Part 6 provides a detailed analysis of the experimental results and part 7 draws conclusions and summarizes the research results.

## 2. THE NATURE OF A DATA MODELING TASK

Previous research suggests that experienced data modelers possess knowledge that falls within four definable categories (Bock & Yager, 2001-2002). The first three of these knowledge categories include learned skills. These are: (1) systems analysis skills, (2) expertise in a data modeling methodology, and (3) software engineering skills. The fourth category is not learned and has to do with an individual's innate cognitive abilities, an inference process. A data modeler must infer how the structure and relationships of items in the database should be modeled based on statements received from users describing required forms and reports (Kroenke, 2004).

> "Reports and forms are like shadows projected on a wall. The users can describe the shadows, but they cannot describe the shapes that give rise to the shadows ... This inferencing process is, unfortunately, more art than science. It is possible to

> learn the tools and techniques for data modeling ... but using those tools and techniques is an art that requires experience guided by intuition" (Kroenke, 2004, pp. 43-44).

The systems analysis skills category refers to a learned set of skills that focus on a designer's ability to extract information about a business problem domain, and to organize the information using different techniques taught in a typical systems analysis and design course (Satzinger, Jackson, & Burd, 2004). Systems analysts are concerned with modeling the flow of business processes along with the data stores used by each business process. In order to model data stores, systems analysts must understand how to use a data modeling methodology (Satzinger, Jackson, & Burd, 2004).

The second category, expertise in a data modeling methodology, refers to one's ability to use one of the well-known data modeling methodologies. The predominant approaches in use today include the ER for relational databases (Rob & Coronel, 2002; Connolly & Begg, 2005) and UML for object-oriented system development (Satzinger, Jackson, & Burd, 2005) diagramming tools. Both the ER and UML methodologies are complex when applied in an operational setting because they require the data modeler to capture characteristics of reality within fairly abstract frameworks. For example, the ER approach abstractly represents an entity such as a customer with a rectangular symbol in a diagram.

The software engineering skills category includes one's learned knowledge with respect to implementation factors that can affect the design of a database model. Theoretically, logical data modeling precedes implementation modeling for a specific database platform; however, in practice the logical data model is rarely developed free from any issues regarding implementation, even though those issues may arise later in the system development life cycle. Even before the logical system design is completed during the analysis stage of the SDLC, technical aspects of hardware, software, and integration requirements must be considered during the planning phase (Rob & Coronel, 2002).

The fourth category is not learned. It has to do with an individual's innate cognitive ability to deal with abstract concepts, specifically abstract modeling concepts. Even though innate ability is not learned, proper training can enable an individual to enhance skills in this category. Unlike rote learning, meaningful learning of abstract concepts is best accomplished by extending the learner's current knowledge with new information that extends what is already internalized (Beishuizen et al., 2002). Ausubel (1963) proposed that instead of memorizing rules, the use of experience and examples allows knowledge extraction by induction; and subsequent research has confirmed this view (e.g., Beishuizen et al., 2002).

In total, the combination of the skills represented by these four categories defines the extent to which an individual can bridge the *cognitive distance* that exists between a modeling task and model solution. The concept of a distance-based categorization model was proposed many years ago by Reed (1972). Estes expanded and tested Reed's model by applying the cognitive distance model to a simulated learning task (1987). According to Estes, "the problem for a decision maker is not only selecting appropriate decision rules, but also one of discriminating among mental representations, with efficiency generally subject to capacity limitations" (Estes, 1987, p. 380). Thus, in the area of data modeling as a type of decision-making task, cognitive distance can be reduced through the use of tools that improve a decision maker's capacity limitations. For example, when instructors teach students techniques that improve on the methodological approach taken to data modeling, the use of techniques that organize data relevant to the decision-making task may reduce the cognitive distance. This research explores the extent to which a simple tool that aids in organizing information about items to be modeled can reduce the cognitive distance for modeling tasks.

## 3. THE DATA MODELING WORKSHEET

Some instructors teach the use of computer-assisted, software-engineering (CASE) tools as part of the database design instructional process. These include complex CASE tools, such as Oracle Corporation's *Designer,* Computer Associates' *ERWin Data Modeler*, or Microsoft's *VISIO* products, as well as simpler data modeling support software, such as *DBDesigner 4* by FabForce.NET, an open source product. Modern database management texts emphasize CASE tools (Date, 2004; Hoffer, Prescott, & McFadden, 2002; Kroenke, 2004).

In addition to the CASE tools noted above, there are numerous other data modeling support tools available with most of the products designed for an operational setting as opposed to classroom use. The DatabaseAnswers.com web site lists 40 data modeling tools that are commercially available. Many of these tools are full-featured and enable the production of all types of diagrams used throughout the software development life cycle including class diagrams, state chart diagrams, activity diagrams, use-case diagrams, and sequence diagrams, among others. Additional features include those that expert systems analysts and software engineers require in order to develop modern information systems productively including drag-and-drop, internationalization language support, DDL and application code generation, and both forward and reverse-engineering of databases, among others. Table 1 provides a description of a selected sample of these support tools.

While CASE tools enable experts to work much more productively than they otherwise could, this does not necessarily apply to novice data modelers, especially students. "In the hands of database novices, CASE tools simply produce impressive-looking bad designs" (Rob & Coronel, 2002, p. 783). Nevertheless, novice data modelers are often taught data modeling concepts while simultaneously learning to use a CASE tool. Developing expertise with CASE tools is difficult because the learning curve is fairly steep (Satzinger, Jackson, & Burd, 2004); thus, teaching a CASE tool while teaching a data modeling methodology can impede the learning process.

| Product/Vendor | Data Modeling Approach | Cost | Comments |
|---|---|---|---|
| Argo UML / *Tigris* | UML | Free (Open Source) | Full-featured diagramming |
| Azzurri Clay / *Azzuri* | ERD | Free | Plug-in to Eclipse |
| ConceptDraw V / *CS Odessa* | ERD, UML | $299 (pro)<br>$149 (std) | Free trial download. |
| DataArchitect / *Sybase* | UML, Comprehensive | $2,000 | Part of *Power Designer* product |
| DDS-Lite (Database Design Studio-Lite) / *Chilli Source* | ERD | $39 (educational) | Professional version available ($299) |
| DBDesigner4 / *FabForce.Net* | ERD | Free( Open Source) | Supports MySQL database integration |
| Dezign / *Datanamic* | ERD | $229 | Generates DDL |
| Enterprise Architect / *Sparx Systems* | Comprehensive UML | $85 to $125 for desktop edition depending on # of licenses ordered. | Covers entire software development lifecycle |
| ER Creator / *modelCreator Software* | ERD | $149 (database edition) | Trial version available including tutorial – database reverse engineering supported |
| AllFusion ERWin Data Modeler / *Computer Associates* | ERD | $3,995 (open license program) | Comprehensive modeling support and code generation |
| MagicDraw / *No Magic Inc.* | UML and Java | $149 (personal edition) | Synchronizes with Eclipse Java libraries |
| Oracle Designer 10g / *Oracle* | ERD | Available through Oracle Academic Initiative (OAI) | Comprehensive support |
| Oracle JDeveloper 10g / *Oracle* | Java modeling and support | Available through OAI | Comprehensive Java development support |
| Poseidon for UML / *GentleWare* | UML | Free (Open Source) community edition | Supports Java forward engineering |
| QDesigner DataArchitect / *Quest Software* | UML + other application modeling methods | $3,895 | Supports all major DBMS products |
| SmartDraw / *SmartDraw* | ERD and UML + others | $148 (technical edition) | One of the easier to learn drawing tools |
| Visio / *Microsoft* | ERD and UML + others | $199 (standard edition) available through Microsoft Academic Alliance | Complete, active in design modeling tool |
| Visual Thought 1.4 / *CERN* | UML + Booch | Free (13-year license) | No longer in production, but free version is available – runs on NT and UNIX |

**Table 1: Sample Support Tools**

As an alternative, we teach students in the introductory database design course to use the data modeling worksheet. The data modeling worksheet is not a computer-based, automated data modeling support tool. It is instead a very simple pencil and paper modeling aid intended to support pedagogical as opposed to commercial efforts.

The data modeling worksheet assists students in organizing information about data entities, data attributes, and relationships among entities. The data modeling worksheet is equally applicable to other modeling methodologies, such as the data modeling component of UML. The data modeling worksheet eliminates the need for novice data modelers to learn a modeling methodology and a CASE tool concurrently. Additionally, CASE tools can be taught later in the course after students have mastered the fundamentals of data modeling. Figure 4 shown later in this article illustrates a completed worksheet for a modeling task.

We contend that an important component of data modeling instruction is the use of a modeling aid or tool that improves a student's ability to deal with abstraction (e.g., Ausubel, 1963; Beishuizen et al., 2002). An approach to data modeling that assists novice designers in organizing facts about a modeling task should help novices make the same orderly transition from one level of problem abstraction to another that efficient modelers utilize (Srinivasan & Te'eni, 1995). This, in turn, narrows the cognitive distance by reducing the number of simultaneous factors with which a novice must contend during the modeling process.

Expert designers are able to categorize constructs (Batra & Davis, 1992) and reduce the complexity of the problem (Srinivasan & Te'eni, 1995) to allow for simplification of the

modeling process. This successive division of a modeling task into smaller subtasks or problems is also the typical approach taken by faculty as they teach data modeling. The data modeling worksheet supports this approach.

The decomposition of a modeling task into sub tasks requires the ability to develop abstract representations of real-world information. In fact, this approach exhibits characteristics associated with pattern-matching approaches to problem solving. As a data modeling task is decomposed, experts tend to match subtasks with a finite number of modeling constructs or patterns available in a given data modeling methodology.

Pattern-matching theory is well documented throughout the cognitive psychology literature. As early as 1972, Reed reported on the importance of pattern recognition and categorization in problem solving. Related to this, Tversky discussed the importance of identifying task features that are similar as a component of problem solving (1977); and Tversky and Gati summarized studies on the categorization of problem tasks based on the identification of similar problem characteristics (1978).

Later, Gick and Holyoak outlined determinants of cognitive transfer and schema induction (1983, 1987). Cognitive transfer has to do with one's ability to transfer problem-solving skills from one domain to another, while schema induction is the development and adoption of different problem-solving approaches. Phye also detailed theory regarding the transfer of analogical reasoning skills with regard to pattern matching (1986, 1990).

The work of these and other cognitive psychologists have laid a foundation over the past several decades that documents the importance of pattern-matching and the development of problem-solving skill sets based on the ability to recognize task features and to categorize those features. Organizing problem domain facts by using an aid, such as the data modeling worksheet, helps formalize the modeling effort. The data modeling worksheet also supports the iterative approach that problem-solving through the use of pattern-matching requires. By providing a formal approach to organize problem domain facts, novices learn to apply problem-solving steps consistently.

## 4. USING THE DATA MODELING WORKSHEET

Consider the following short extract from a modeling problem that may be presented to students as part of an introductory data modeling course to illustrate use of the data modeling worksheet.

> ABC Company has 11 different **departments**, and each has a unique name. 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 and meeting comments between a department and a vendor.

Experienced data modelers tend to analyze this type of problem description by decomposing it a sentence or two at a time. This enables the identification of potential entities, which we have indicated in bold print, and their associated attributes, which we have underscored in the problem statement. Relationships among entities are often represented by verbs in a textual, problem description, and are identified by iterating through a problem description.

As entities, attributes, and relationships are identified, these domain facts are recorded on the data modeling worksheet. The data modeling worksheet is divided into two columns, as illustrated in Figure 1. Students record facts about entities and attributes in the left-side column. The right-side column is used to draw simple depictions of individual relationships as these relationships are identified. For example, after reading one or two sentences in the problem described above, students may identify the potential existence of DEPARTMENT, VENDOR, and EQUIPMENT entities and their associated attributes. Students record the information on the data modeling worksheet in the left-side column as shown in Figure 1. Attributes that are candidate identifiers are underlined.

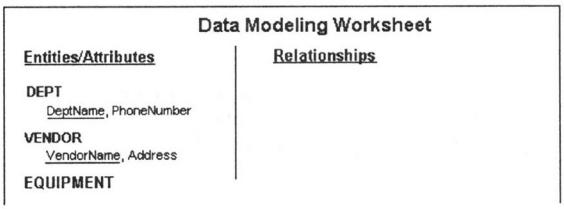



**Figure 1: Initial Data Modeling Worksheet**

It is important to emphasize that any relationship depictions must be drawn as only individual relationships. At this point in the modeling effort, students should not attempt to combine relationships into an integrated solution diagram of the modeling task. Doing so can hinder the efforts made to organize the domain information that is relevant to the task.

Different modeling approaches use different techniques for recording information about relationships among entities or objects. The ER modeling methodology has four basic relationship patterns—unary, binary, ternary, and the generalization hierarchy. These are described in Table 2 and illustrated in Figure 2. These relationship patterns comprise the vast majority of existing business relationships. Within these four basic patterns, there are pattern variations, such as the strong entity-weak entity binary relationship. Students learn to match modeling subtask characteristics to the basic modeling patterns and their variations.

Returning to our problem statement, a student should detect the existence a relationship between DEPARTMENT and VENDOR as indicated by the phrase "department *deals with* many vendors." Students record this information by drawing a depiction of the binary relationship in the right-side column

of the data modeling worksheet as illustrated in Figure 3. When sufficient information is available, students may also identify the maximum cardinality of the relationship (many-to-many for this binary relationship) as part of an individual relationship diagram.

| | |
|---|---|
| Unary Relationship | a relationship between instances of an entity and other instances of the same entity. |
| Binary Relationship | a relationship between instances of one entity and instances of a second entity. |
| Ternary Relationship | a relationship among instances of three entities. |
| Generalization Hierarchy | a categorization of classes wherein an instance of a superclass is related to instances of one or more subclasses. |

**Table 2: Basic Relationship Patterns**

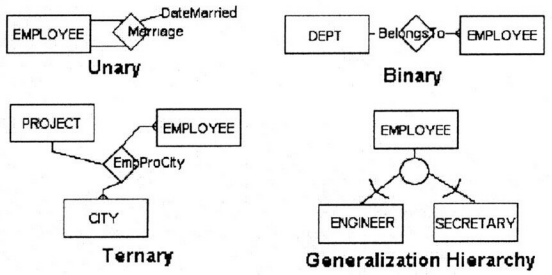



**Figure 2: Basic Relationship Patterns**

Additionally, there are *association* attributes (also termed *intersection* attributes) that are a function of the relationship, and these are also recorded in the right-side column as part of the relationship depictions. In a relational implementation, association attributes are determined by the primary keys of each entity that participates in the relationship. The *date of meeting* and meeting *comments* represent important information to be stored in the database because managers may use this information to make future decisions. Association attributes are depicted as part of the relationship drawings in the right-side column of the worksheet as shown for the *DealsWith* relationship in Figure 3.

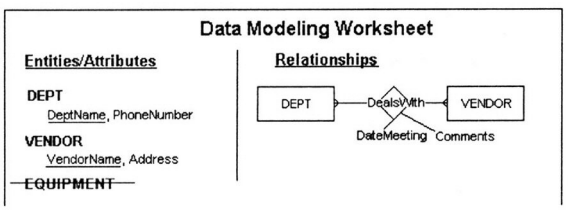



**Figure 3: Revised Data Modeling Worksheet**

In most modeling tasks, designers will identify potential entities that will eventually be eliminated in the final solution. One reason for not modeling an entity is that it has no attributes about which the firm needs to record data. This is the situation for the EQUIPMENT entity initially identified in Figure 1. Once a student determines that an entity need not be included in the final solution, they can simply line-through the entity as is done for the EQUIPMENT entity in Figure 4.

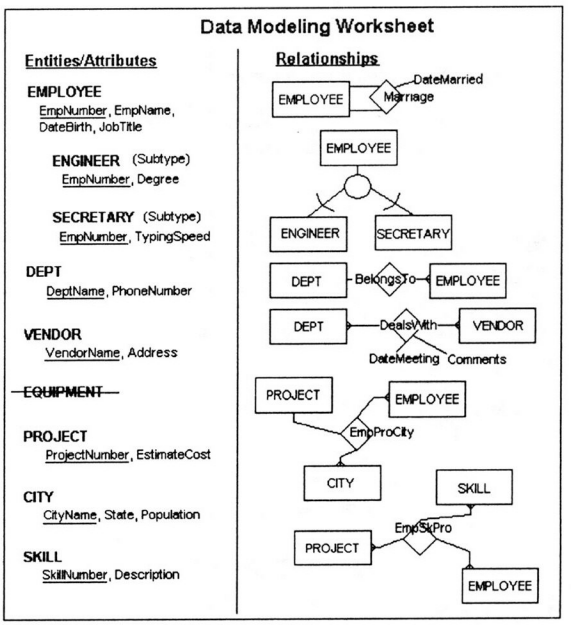



**Figure 4: Complete Data Modeling Worksheet**

Upon completing the analysis, the data modeling worksheet will contain all of the information students need to complete the modeling exercise in an organized fashion. Figure 4 illustrates a completed data modeling worksheet for the modeling task described in Appendix 1.

The information in the completed data modeling worksheet can be integrated into an overall model solution diagram, if that is the goal instructors set as an objective of the modeling exercise. This will enable students to depict all relevant entities, attributes, and relationships graphically. Figure 5 illustrates the overall solution diagram.

Contrast this modeling approach to that typically taken by students who are not taught a method for organizing the information to be modeled. Our experience has shown that it is very typical for students to begin to solve a modeling problem by immediately proceeding to the design (drawing) of a single, large modeling diagram like that illustrated in Figure 5. Without an aid for organizing problem domain facts, novice students often experience difficulty in arriving at a correct solution. A typical error is a failure to distinguish between entities to be modeled and "objects" that are not actual entities. For example, many novices will model *ABC Company* (the firm's name) as an entity in their solution diagram. The resulting solution diagram may also exhibit errors because the students have not developed an adequate understanding of the problem domain.

Iteration is also difficult with the "large diagram" approach as students iterate by erasing and redrawing portions of the modeling diagram. Supporting an iterative problem-solving

approach is one of the strengths of the data modeling worksheet approach. In the next section, we detail the research methodology used to test the hypothesis that students can benefit from the use of an organization tool that reduces the cognitive distance associated with the modeling problem.

**EER Solution to the Experimental Modeling Task**

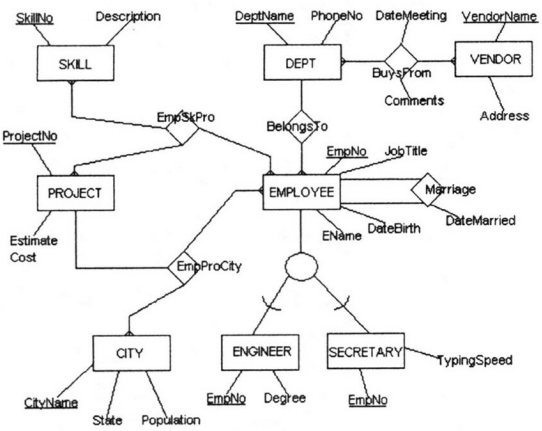



**Figure 5: EER Solution**

## 5. RESEARCH METHODOLOGY

### 5.1 Research Design

A laboratory experiment compared the performance for two groups of students which are denoted as *1-Worksheet* (n=96) and *2-No Worksheet* (n-57). Group 1 was taught data modeling with the data modeling worksheet while Group 2 received comparable instruction with no emphasis on how to organize information contained in a problem statement. The experimental design is a derivation of the post-test-only control group design. The 153 observations provide error probabilities of $\alpha = 0.05$ and $\beta < 0.10$ that the detection of treatment effects will not be detected if significant differences exist between the performance of the two groups.

### 5.2 Experimental Variables and Performance Evaluation

*Model correctness* is the dependent variable and is measured by the degree of correctness in a subject's final model diagram. We used the *facet* measurement approach and grading scheme for evaluating modeling correctness used by Batra, Hoffer, & Bostrom (1990). The facets for this experiment are the correct modeling of: (1) entities, (2) entity identifiers, (3) a unary relationship, (4) a binary one-to-many relationship, (5) a binary many-to-many relationship, (6) a ternary one-to-many-to-many relationship, (7) a ternary many-to-many-to-many relationship, and (8) a generalization hierarchy. This measurement approach is more valid than one that attempts to develop an overall correctness measure for the model because such a measure would lack construct validity. Measuring correctness at the facet level is also a very intuitive approach because database models are not either correct or incorrect; rather, they have different degrees of correctness. Table 3 identifies the facets used in evaluating correctness.

| Facet | Evaluation Criteria |
|---|---|
| Entity | Entity is properly identified. |
| Identifier | A key identifier attribute is modeled for each entity. |
| Unary 1:1 Relationship | Modeled properly. |
| Binary 1:M Relationship | Modeled properly. |
| Binary M:N | Modeled properly. |
| Ternary 1:M:N | Modeled properly. |
| Ternary M: N: O | Modeled properly. |
| Category (Generalization Hierarchy) | Modeled properly. |

**Table 3: Modeling Facets**

Table 4 identifies the grading scheme for evaluating facets. Each facet is graded separately with a score of 1 for a correct facet and 0 for an incorrect facet. The protocol specifies two classes of intermediate errors: *medium* and *minor*. Scores awarded for facets with medium and minor errors are 0.50 and 0.75 points, respectively.

Where errors were made that were not in the grading scheme, the grader subjectively evaluated the errors according to their effect on the data model's ability to capture the semantic meaning of the data. Individual facet scores were converted to a percentage score (zero to 100 percent).

Control variables in the research include *characteristics of the task*, which are held constant through use of a single experimental task, and *characteristics of the human subject*, which are controlled through the use of large group sizes. We were concerned with the application of individual *problem-solving schemata* based on individual learned knowledge as described by Gick and Holyoak (1983). Problem-solving schemata is the term used in cognitive psychology to refer to the problem-solving approach that individuals develop through experience and training over time. The experiment controlled the development of individual schemata with respect to the data modeling approach learned by subjects, but did not control for an individual's overall general schemata based on prior problem-solving learning and experience. In general, it appears that this latter concern should be adequately controlled through the randomization of human characteristics resulting from large group sizes for the two groups; however, we have no means for ensuring this control.

The subjects came from two undergraduate, senior-level college sections of the same course on database management systems. All subjects selected had previously completed prerequisites for the course and had not received prior instruction in the data modeling methodology. All of the subjects had completed a course that covered process modeling. Students with prior data modeling experience were eliminated from the subject pool.

| Facet | Incorrect | Medium Error | Minor Error |
|---|---|---|---|
| Entity | • Missing<br>• Represented as attribute instead of as a relationship | • | • Modeled extra entity |
| Identifier | • Missing<br>• Identifier is different from that in the task | • -Not underlined | |
| Relationship | • Missing<br>• Incorrect Degree<br>• Modeled as generalization hierarchy | • Incorrect connectivity<br>• Unary relationship modeled by categories | • Unary relationship modeled as attribute |
| Generalization Hierarchy | • Missing | • Incorrect representation by using a relationship symbol | |

**Table 4: Error Classification Grading Scheme**

### 5.1 Hypotheses

Eight hypotheses were tested, each corresponding to a facet of the modeling task. In the null form, there is no significant difference in model correctness for subjects in the 1-Worksheet and 2-No Worksheet groups for eight different facets:

- $H_1$-Entities
- $H_2$-Identifier Attributes
- $H_3$-Unary One-to-One Relationships
- $H_4$-Binary One-to-Many Relationships
- $H_5$-Binary Many-to-Many Relationships
- $H_6$-Ternary One-to-Many-to-Many Relationships
- $H_7$-Ternary Many-to-Many-to-Many Relationships
- $H_8$-Generalization Hierarchies.

### 5.4. Experimental Procedure

Each group was trained following a standard eight-hour curriculum that included consecutive instructional periods over a three-week timeframe as outlined in Table 5. The block of instruction is a standardized eight-hour curriculum developed in a prior pilot study. The data modeling methodology taught was the Extended Entity-Relationship (EER) model. The EER model extends the original ER model by adding a modeling facet for generalization hierarchies, also referred to as categories (Elmasri, Weeldreyer, & Hevner, 1985).

The same instructor taught both groups and followed a planned note set. The treatment group was trained with the data modeling worksheet by incorporating the use of worksheets in the instruction. The example modeling problems were identical for each group, and neither group exhibited serious mortality. In order to stimulate subject motivation, subjects were graded and received credit for their performance as part of their course work; however, students had the option to opt out of the study. Students opting out of the study were not included in the subject groups. Proficiency in the modeling method and use of the worksheet was not evaluated prior to the completion of the experimental modeling task.

Following the instruction, each subject completed a modeling exercise previously used by Batra, Hoffer, and Bostrom (1990) as described in Appendix 1 and illustrated earlier in Figure 5. Subjects were provided a maximum of 75 minutes to complete the task with most subjects finishing in an average of 46 minutes.

| Topic | Time Allocated |
|---|---|
| Basic concepts about data, entities, relationships and association data. | 45 minutes |
| Diagrammatical modeling with the EER method | 45 minutes |
| Applying EER modeling to simple and complex data relationships with examples of unary, binary, and ternary relationships | 3 hours, 30 minutes |
| Data subclasses, superclasses, and generalization | 1 hour |
| Practice modeling exercises | 2 hours |
| TOTAL | 8 hours |

**Table 5: Training Curriculum**

## 6. ANALYSIS AND RESULTS

The analysis approach is a one-way ANOVA for each facet. While the experiment-wise statistical significance level was set at $\alpha = 0.05$, the fact that multiple dependent comparisons were made for the same subjects required a reduction in the significance level used for individual comparisons in accordance with the Bonferroni inequality. For those unfamiliar with this approach, reducing the significance level for rejection of the null hypotheses is appropriate for multiple, dependent comparisons in order to avoid rejecting a null hypothesis out of mere chance. The reduced significance level computed to an F-test significance level of $\alpha = 0.006$ for each of the hypotheses tested. A power analysis showed a $\beta < 0.10$ as planned to avoid Type II errors; thus, the sample sizes for the two groups were sufficiently large for the research.

Table 6 details the results of the ANOVA. Significant improvements in modeling performance were found for H1, H2, H6, and H7. Use of the data modeling worksheet improved subject performance in the identification of both entities and their identifier attributes correctly ($p < 0.001$).

| Facet | 1-Worksheet Mean / Std Dev. (N = 96) | 2-No Worksheet Mean / Std Dev. (N = 57) | F-statistic | p-value |
|---|---|---|---|---|
| $H_1$-Entity | 97.3 5.5 | 91.5 8.9 | 25.657 | **0.000** |
| $H_2$-Identifier | 95.3 10.3 | 82.2 18.8 | 31.038 | **0.000** |
| $H_3$-Unary one-one | 93.5 18.9 | 85.3 25.7 | 5.092 | 0.025 NS |
| $H_4$-Binary one-many | 89.6 21.7 | 87.1 25.8 | 0.422 | 0.517 NS |
| $H_5$-Binary many-many | 92.7 20.5 | 98.7 7.3 | 4.607 | 0.033 NS |
| $H_6$-Ternary one-many-many | 46.9 26.0 | 12.1 23.5 | 69.632 | **0.000** |
| $H_7$-Ternary many-many-many | 63.0 37.9 | 12.1 27.0 | 80.255 | **0.000** |
| $H_8$-Generalization Hierarchy | 84.9 27.3 | 75.9 26.9 | 4.011 | 0.047 NS |

**Table 6: Results**

Group 1 also showed significantly better performance in modeling both types of ternary relationships ($p < 0.001$). This is an important finding, as it is generally recognized that ternary relationships are the most difficult relationships for students to learn to model correctly.

No significant difference in modeling performance was found for modeling unary and binary relationships or a generalization hierarchy. However, Table 6 indicates that the percentage of correctness for H3, H4, H5 were quite high. This is consistent with other research (e.g., Amer, 1993; Batra, Hoffer, & Bostrom, 1990; Liao & Palvia, 2000) that has shown binary and unary relationships to be the easiest types of relationships to model. The statistical analysis does not find differences in the two groups for these hypotheses because of the ceiling effect associated with measuring performance as a percentage. As the correctness level approaches 100%, a significant difference in the variance of the groups will not be detected.

There was also no detectable significant difference in modeling the generalization hierarchy in the experimental task. The Worksheet group scored 84.9% correctness with a variance of 27.3 on this facet while the No Worksheet group scored 75.9% with a variance of 26.9. Modeling a generalization hierarchy is very closely related to the task of modeling a binary relationship. In fact, the physical implementation of a generalization hierarchy is often a binary relationship. We conclude use of the data modeling worksheet does not significantly affect the ability to recognize a generalization hierarchy.

## 7. CONCLUSIONS AND SUMMARY

The results of the research support the overall hypothesis that the data modeling worksheet assists in reducing the cognitive distance between a modeling task and a modeling solution. Four of the facets evaluated showed statistically significant improvements in modeling performance with no significant differences in performance for the additional facets that were evaluated.

One of the exciting results when using the data modeling worksheet is the performance improvement for modeling ternary relationships. A total of 47% of the subjects that used the data modeling worksheet modeled the ternary one-many-many relationship correctly compared to only 12% for the subjects that did not use the worksheet. For the ternary many-many-many relationship, a total of 63% of the subjects that used the data modeling worksheet modeled the relationship correctly compared to only 12% for those not using the worksheet.

We believe that the superior performance of the subject group that used the data modeling worksheet resulted from an improved ability to organize information as it was extracted from the modeling task. This is a testable hypothesis and deserves additional study. A useful approach would be to extend a replication of this current research to include a visual recording of subject modeling efforts. While such a research design would be expensive and time-consuming, it would facilitate a meta analysis of the approaches that subjects take when extracting and organizing task information.

Another extension of the research could focus on task complexity variation. The task in this research involved six entities with two additional subclass entities and six relationships. This limitation in the research was an artifact of using the classroom as a research laboratory and the time restrictions that are inherent in this approach. It would be interesting to observe differences in performance for a set of increasingly complex tasks, perhaps up to the point that the task domain might have 40 or more entities with 30 to 50 relationships.

One of the limitations of this study is that the individual subjects were not trained to a specific standard prior to administering the experimental task. The two experimental groups did receive comparable eight-hour training sessions; however, this does not guarantee that each subject achieved the same degree of competence with the EER modeling methodology. In our opinion this particular limitation has minimal impact on the validity of the findings because of the results outlined in Table 6. Note that the percentage of correctness achieved for $H_1$, $H_2$, $H_3$, $H_4$, $H_5$, and $H_8$ were all quite high. This indicates that the level of learning was sufficiently high to add validity to the findings. Additionally, the large group sizes should normalize the distribution of this potential confounding error.

We conclude by underscoring two of our earlier points. First, the data modeling worksheet can be used to teach

different modeling methodologies. While this research used the EER model, the data modeling worksheet can be easily adapted to other data modeling methodologies. Second, while we believe that the use of CASE tools during instruction on data modeling methodologies can hinder learning the modeling methodology, this does not preclude their introduction later in a course of study after students have learned basic data modeling concepts. We did not empirically test our presumption that CASE tools may hinder learning, and this question also deserves additional study.

It is also important to understand that the data modeling worksheet provides a method for formalizing the normal, iterative problem-solving process. The data modeling worksheet is a very easy tool for students to learn and use. It clearly reduces the cognitive distance in the modeling effort for novice students.

## 8. REFERENCES


Amer, T., "Entity-Relationship and Relational Database Modeling Representations for the Audit Review of Accounting Applications: An Experimental Examination of Effectiveness." Journal of Information Systems, Vol. 7, No. 1, 1993, pp. 1-15.

Ausubel, D (1963), The Psychology of Meaningful Verbal Learning, Grune & Stratton, New York, NY.

Batra, D., and J. G. Davis, "Conceptual Data Modeling In Database Design – Similarities and Differences between Expert and Novice Designers." International Journal of Man-0Machine Studies, Vol. 37, No. 1, 1992, pp. 83-101.

Batra, D., J. A. Hoffer, and R. P. Bostrom, "Comparing Representations Developed Using Relational and EER Models." Communications of the ACM, Vol. 33, No. 22, 1990, pp. 126-139.

Beishuizen, J., E. Stoutjesdijk, S. Spuijbroek, S. Bouwmeester, and H. van der Geest, "Understanding Abstract Expository Texts." British Journal of Educational Psychology, Vol. 72, 2002, pp. 279-297.

Bock, D. B. and S. E. and Yager, "Improving Entity Relationship Modeling Accuracy with Novice Data Modelers." Journal of Computer Information Systems, Vol. 42, No. 2, 2001-02, pp. 69-75.

Connolly, T. M. and C. E. Begg (2005), Database Systems: A Practical Approach to Design, Implementation, and Management, 4th edition, Addison-Wesley, Boston, MA, Chapter 9.

DatabaseAnswers.Com, Retrieved May 2005 from http://www.databaseanswers.com/modelling_tools.htm.

Date, C. J. (2004), An Introduction to Database Systems, 8th Edition, Addison-Wesley, Boston, MA, Chapters 14, 20, and 25.

Elmasri, R., J. Weeldreyer, and A. Hevner. "The Category Concept: An Extension to the Entity-Relationship Model," Data Knowledge Engineering, Vol. 1, No. 11, 1985, pp. 75-116.

Estes, W. K. "Application of a Cognitive-Distance Model to Learning in a Simulated Travel Task," Journal of Experimental Psychology, Vol. 13, No. 3, 1987, pp. 380-386.

Gick, M. L. and K. J. Holyoak. "Schema Induction and Analogical Transfer." Cognitive Psychology, Vol. 15, 1983, pp. 1-38.

Gick, M. L. and K. J. Holyoak (1987), "The Cognitive Basis of Knowledge Transfer" in Transfer of Learning: Contemporary Research, by Cormier, S. M. and J. D. Hagman, Editors, Academic Press, Chapter 3.

Hoffer, J. A., M. B. Prescott, and F. R. McFadden (2002), Modern Database Management, 6th Edition, Prentice-Hall, Upper Saddle River, New Jersey, Chapters 3, 4, and 14.

Konar, A. (2000), Artificial intelligence and soft computing: Behavioral and Cognitive Modeling of the Human Brain, CRC Press, Boca Raton, FL, Chapter 2.

Kroenke, D. (2004), Database Processing: Fundamentals, Design, and Implementation, 8th Edition, Prentice-Hall, Upper Saddle River, New Jersey, Chapters 2, 3, and 16.

Liao, C. C. and P. C. Palvia, "The Impact of Data Models and Task Complexity on End-User Performance: An Experimental Investigation." International Journal of Human-Computer Studies, Vol. 52, No. 5, 2000, pp. 831-845.

Oracle Designer 10g, Oracle Technology Network, retrieved July 2004, from http://www.oracle.com/technology/products/designer/index.html.

Phye, G. D. (1986), Transfer of Analogical Reasoning Skills. American Educational Research Association, San Francisco, California, Chapter 1.

Phye, G. D., "Inductive Problem Solving: Schema Inducement and Memory-Based Transfer." Journal of Educational Psychology, Vol. 82, No. 4, 1990, pp. 826-831.

Reed, S. K., "Pattern Recognition and Categorization." Cognitive Psychology, Vol. 3, 1972, pp. 382-407.

Rob, P. and C. Coronel (2002), Database Systems: Design, Implementation, and Management, 5th edition, Course Technology, Boston, MA, Chapters 3, 6, and 16.

Satzinger, J. W., R. B. Jackson, and S. D. Burd (2004), Systems Analysis and Design in a Changing World, 3rd edition, Course Technology, Boston, MA, Chapters 1, 2, and 16.

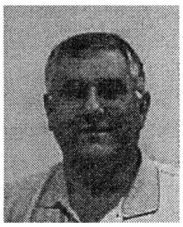
Satzinger, J. W., R. B. Jackson, and S. D. Burd (2005), Object-Oriented Analysis and Design with the Unified Process, Course Technology, Boston, MA, Chapter 2.

Srinivasan, A. and D. Te'eni, "Modeling As Constrained Problem Solving: An Empirical Study of the Data Modeling Process." Management Science, Vol. 41, No. 3, 1995, pp. 419-434.

Tversky, A. "Features of Similarity." Psychological Review, Vol. 84, 1977, pp. 327-352.

Tversky, A. and I. Gati (1978), "Studies of Similarity" in Cognition and Categorization, Rosch, E., and Lloyd, B. B., Editors, pp. 79-98.

## AUTHOR BIOGRAPHIES



**Douglas B. Bock** is a Professor in Computer Management and Information Systems in the School of Business, Southern Illinois University Edwardsville. His primary teaching area is database management systems including database modeling, implementation, administration, and programming using Microsoft's .NET framework. His current scholarship focuses on the development of pedagogical materials, primarily textbooks. He recently coauthored two textbooks with Dr. Bijoy Bordoloi, *Oracle SQL* (2004, also published in Dutch under the title *Sql Voor Het Hoger Onderwijs*) and *SQL for SQL Server* (2004), both published by Prentice-Hall Publishing Company. His research addresses how information systems professionals model the data used by managers in decision making situations. Dr. Bock has published and presented over 40 papers in professional journals and at professional conferences, including *Decision Sciences*, *Communications of the ACM*, *Journal of Systems and Software*, *Journal of Computer Information Systems*, and *Journal of Database Management*, among others. His Ph.D. is in Management Information Systems from Indiana University (1987), and he is a Microsoft Certified Professional (MCSD.NET).

**Susan E. Yager** is an Associate Professor in Computer Management and Information Systems at Southern Illinois University Edwardsville. She holds a Ph.D. in Business Computer Information Systems. Her recent teaching responsibilities include introduction to MIS, database design, and RAD. Susan's research interests focus on technology use within organizations, especially the use of virtual teams and the impacts of information technology acquisition, and the evaluation of technology use in business school curriculums. She has published work in *Communications of the ACM* plus several of ACM's special interest groups, *Journal of Computer Information Systems*, *Journal of Education for Business*, *Dispute Resolution Journal*, and proceedings of national and regional conferences.

## APPENDIX I. Narrative Description of the Experimental Task.

(Batra, Hoffer, and Bostrom, 1990)

Study the modeling problem described below and draw a diagram that represents a conceptual model of the problem situation. Do not create data attributes not mentioned in the problem situation. Use the attached sheet to draw your final solution. Use the scratch paper handed out to draw rough drafts of the solution. After completing this task, you will be asked to complete a short questionnaire concerning your computer-related skills and experience and the difficulty of this modeling task.

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 and meeting comments 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 city name, state name, 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.