Association for Information Systems
AIS Electronic Library (AISeL)

SAIS 2005 Proceedings

Southern (SAIS)

3-1-2005

Online Data Modeling to Improve Students' Learning of Conceptual Data Modeling

Hsiang-Jui Kung
hjkung@georgiasouthern.edu

Hui-Lien Tung

Recommended Citation

Kung, Hsiang-Jui and Tung, Hui-Lien, "Online Data Modeling to Improve Students' Learning of Conceptual Data Modeling " (2005). *SAIS 2005 Proceedings*. 28.
http://aisel.aisnet.org/sais2005/28

ONLINE DATA MODELING TOOL TO IMPROVE STUDENTS' LEARNING OF CONCEPTUAL DATA MODELING

Hsiang-Jui Kung
Georgia Southern University
hjkung@georgiasouthern.edu

Hui-Lien Tung
Troy State University
htung@troyst.edu

Abstract

Traditionally, IS/IT educators teach data modeling by using top-down approach that starts from business rules, entity-relationship (E-R) diagram then normalization and ends with normalized E-R diagram. Another approach is bottom-up that analyzes business documents and then develop a data model. A laboratory study will be conducted that employed IS/IT students (novice designers) to compare the effectiveness of the two design approaches (between subjects) at three levels of task complexity (within subjects). The research design includes the conceptual framework, experimental design and sample tasks. Both design approaches will be covered in the three sections of Systems Analysis and Design course but with different sequence to eliminate bias.

Keywords: Database Design, Data Modeling, Normalization

Introduction

Conceptual data modeling is one of the major phases of systems development (Hoffer, George and Valacich, 2004, Batra and Wishart, 2004). Data modeling is a problem solving activity. Usually novice designers make errors in this phase and consider the task difficult. Errors in this phase lead to poor database quality that may generate inaccurate results and degrade business decisions (Martin and Leban, 1995). Entity-relationship diagram (Chen, 1976) is the most widely used form for data model. CASE tools have been introduced trying to decrease data modeling errors but produced little help since CASE tools can not detect data modeling errors.

Normalization is the most common database analysis technique to validate data model. Database designers usually normalize the database to the third normal form (3NF) to eliminate insertion, deletion and updating anomalies and reduce data redundancy (Codd, 1970). Top-down design approach integrates data modeling with normalization. This approach first derives the business rules for a data model from an intimate understanding of the nature of the business. The business rules are gathered from a series of questions to identify entities, primary key(s) and attributes, relationships and their cardinality and degree. The second step is to normalize the data model using functional dependencies. The last step is to revise the data model from the first step based on the results of the normalization process. Most textbooks use this approach to illustrate database conceptual design (Hoffer et al, 2002, Satzinger, Jackson and Burd, 2002, Whitten, Bentley and Dittman, 2004).

Bottom-up design approach gathers the information for data modeling by analyzing specific business documents—computer displays, reports, and business forms—handled within the system. This approach gains the understanding of the data that must be maintained in the system's database. The initial data can be treated as a big table/relation. The designers decompose the big table into several smaller tables/relations in 3NF which is called the normalization process. The designers then translate the relational model to entity-relationship diagram. This approach first defines attributes and then groups them to form entities (Rob and Coronel, 2004). The selection of a design approach often depends on the data availability and personal preferences. Because there is no one approaches for all situations, a designer must be proficient in both approaches so he or she can handle different situations.

Previous studies have shown that the primary cause of novice designer performance weakening is due to relationships between increased entities, and not because of associations among entities and attributes (Batra and Antony, 1994, Shoval and Shiran, 1997). For example, with five interrelated entities, the number of possible binary relationships is 24 (4×3×2). Relationship is not the only factor contributing to data modeling complexity. Cardinality is another factor making data modeling more complicated. The number of possible binary relationships for the same five entities cardinalities is 192 (4×3×2×8). Among the large set of relationships and cardinalities, only 3 or 4 will be the correct solution. Adding attributes and primary keys to the data model, some novice designers may totally mess up the data model.

In this study, we will compare top-down and bottom-up conceptual database design approaches at three levels of task complexity. Subjects of this study are students enrolled in three sections of systems analysis and design course. A subject will use both approaches and work at all three complexity levels. The number of errors in terms of entities, attributes, primary keys, relationships and cardinalities will be measured and analyzed.

Research Framework and Hypotheses

The research framework (see Figure 1) shows the variables used in this study. The dependent variable is the designer performance (the number of errors divided by the total number of objects: entity, attribute, primary key, relationship and cardinality). The lower error rate indicates the higher designer performance. The rationale of the use of error rate instead of score is to provide a mean to identify the designers' problem areas directly. Designer performance depends upon the design approach used (top-down vs. bottom-up) and the task complexity (low, medium and high). Other demographic factors, academic status, gender, major, prior database related courses, will be collected and tested.

Since all the subjects will use both approaches, the sequence of using the two approaches may affect designers' performance. We will randomize the sequence to minimize the bias. Our main interest is finding the performance differences between two design approaches and at three complexity levels. Our hypotheses are as follows:

H1. The design performance remains the same regardless of complexity levels.

Since both design approaches have their strength and weakness in different situation, we also expect an interaction effect between design approach and complexity level.

H2. There will be an interaction effect between design approach and complexity level.

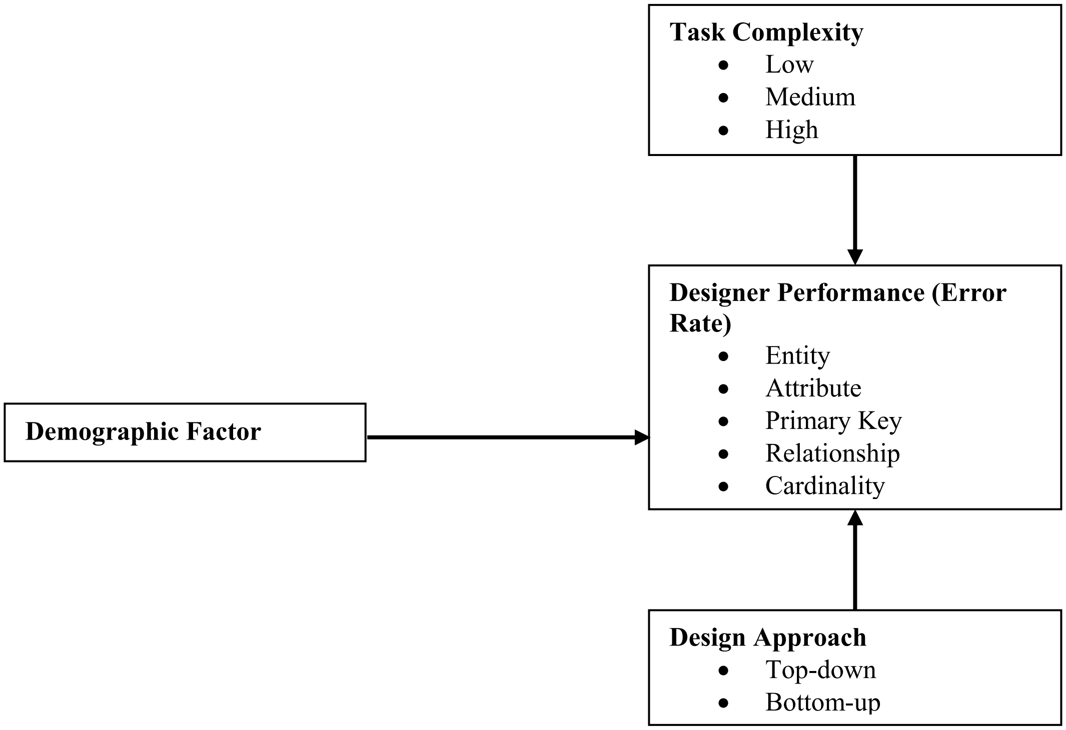
H3. Design performance remains the same regardless of design approaches.

Finally, we would like to identify the problem areas for novice designers when applying the two design approaches at three complexity level. We assume that error rate in all the categories remain the same for both approaches at all complexity levels.

H4. Design performance remains the same in all the categories for both design approaches at all three complexity levels.

We would like to find out which categories have higher error rate using which design approach and at what complexity level. The findings will be the problem areas for novice designer. This information can help IS educators to allocate their resources (time) targeting those problem areas in order to help students to improve their data modeling skill.

Figure 1. Research framework



Research Methodology

This study will be conducted as a laboratory experiment. To represent novice database designers, about 60 students enrolled in both undergraduate and graduate systems analysis and design courses (three sections) will be recruited. Students' demographic factors: gender, major, academic status, prior database related courses will be recorded.

Procedure

Students will have ten 75-minutes sessions. In the first six sessions, students in each section will be exposed to data modeling, relational model, normalization and integration. The last four sessions formed the research experiment.

In the first session, the instructor explains fundamental database concepts and terminologies. The conceptual data modeling is introduced in the second session. The relational model is covered in the third session. The importance of database normalization and three normal forms are covered in the fourth session. The normalization technique including examples is demonstrated in the fifth session. In the sixth session, students will learn how to combine ERD with normalization. According to the randomly selected order, the students will have problems to practice using the two design approaches at all three complexity levels, one session at a time. For example, if the order selected is first top-down then bottom-up, the students will use top-down approach to solve problems in the seventh session. Then use the bottom-up approach in the eighth session. Using the same sequence, students will solve different problems in the ninth and tenth sessions. Results from these last two sessions will be evaluated for this study.

Students may perform better in the second design approach since they gained experience in solving data modeling problems. The random application sequence will minimize the bias in favor of one approach over the other.

Task Complexity

Low complexity: This task has three entities and two binary relationships. Every entity has some attributes and one of the attribute is the primary key.

Medium complexity: This task has five entities (one of the entity is an associative entity) and four binary relationships. Every entity has some attributes and the associative entity has a composite key.

High complexity: This task has seven entities (one of the entity is an associative entity) and six relationships (one of the relationships is ternary). Every entity has some attributes and the associative entity has a composite key.

Design Approach

The sample problems for the two design approaches at the medium complexity level are as follow:

1. Top-down design task:
   You are assigned to design a database for a university. You need to develop a data model using top-down design approach.

   1) You need to draw initial E-R diagram according to the business rules provided by your clients as follows:
      A course may have many classes and a class is related to only one course. A student may take many classes and a class should have at least 10 students enrolled to. A professor may teach many classes and a class is taught by only one professor.

   2) An experienced DBA observes the sample data and identities functional dependencies. You need to normalize the relation to 3NF and redraw your E-R diagram with attributes.
      Course# → CourseName

      Student# → StudentName

      Student#, Class# → Grade, Course#, CourseName, StudentName, Credit, ProfessorName, Sect., Prof#

      Prof# → ProfessorName

      Class# → Course#, CourseName, Room#, Prof#, Sect., ProfessorName, Credit

2. Bottom-up design task:

   You are assigned to design a database for a Tool Tracking System. An experienced DBA identifies a relation ToolSystem and functional dependencies for you. Your task is to normalize the relation into the third normal form, and draw an E-R diagram.

   ToolSystem(Emp#, EmpName, EmpPhone#, Dept#, DeptName, DeptLocation, <u>CheckOut#</u>, CheckOutDate, <u>Line#</u>, Tool#, ToolName, Quantity)

   Functional Dependencies:

   Emp# → EmpName, EmpPhone#, Dept#, DeptName, DeptLocation

   Dept# →DeptName, DeptLocation

   CheckOut# → Emp#, CheckOutDate, EmpPhone#, Dept#, DeptName, DeptLocation

   CheckOut#, Line# → Emp#, CheckOutDate, EmpPhone#, Dept#, DeptName, DeptLocation, Tool#, ToolName, Quantity

   Tool# → ToolName

Design Performance (Error Rate)

Design performance is measured by the number of errors in all the categories: entity, attribute, primary key, relationship and cardinality, and then divided by the total number of objects in that category. The overall error rate is the sum of all the errors divided by the sum of all the objects in every category. The performance marking scheme is fairly algorithmic in nature by counting how many error types (see Table 1) and then divided by total number of objects.

Table 1. Error Type

Object	Error Type

Entity	Extra entity	Missing entity	
Attribute	Missing	Extra attribute	
Primary Key	Wrong key	Missing key	Extra key
Relationship	Degree error	Missing connectivity	Extra relationship
Cardinality	Wrong	Missing	

References

Batra, D., & Antony, S. R. (1994) Novice errors in database design. *European Journal of Information Systems*, 3 (1), 57-69.

Batra, D., & Wishart, N. A. (2004) Comparing a rule-based approach with a pattern-based approach at different levels of complexity of conceptual data modeling tasks. *International Journal of Human-Computer Studies*, 61, 397-419.

Chen, P. P. (1976) The entity-relationship model—toward a unified view of data. *ACM Transactions on Database Systems*, 1 (1), 9-36.

Codd, E. F. (1970) A relational model of data for large relational databases. *Communications of the ACM*, 13 (June), 377-387.

Hoffer, J. A., George, J. F., & Valacich, J. S. (2002) *Modern Systems Analysis and Design*. Prentice-Hall, Upper Saddle River, NJ.

Martin, J., & Leben, J. (1995) *Cleint/Server Databases: Enterprise Computing*. Prentice-Hall, Upper Saddle River, NJ.

Rob, P., & Coronel, C. (2004) *Database Systems: Design, Implementation, and Management*. Course Technology, Boston, MA.

Satzinger, J. W., Jackson, R. B., & Burd, S. D. (2002) *Systems Analysis and Design*. Course Technology, Boston, MA.

Shoval, P., & Shiran, S. (1997) Entity-relationship and object-oriented data modeling—An experimental comparison of design quality. *Data and Knowledge Engineering*, 21 (3), 297-315.

Whitten, J. L., Bentley, L. D., & Dittman, K. C. (2004) *Systems Analysis and Design Methods*. McGraw-Hill, New York, NY.