A Review of the First Phase of a Project to Develop and Utilize Animated Database Courseware

Meg Murray
mcmurray@kennesaw.edu

Mario Guimaraes

Follow this and additional works at: http://aisel.aisnet.org/sais2006

Recommended Citation
http://aisel.aisnet.org/sais2006/44
A REVIEW OF THE FIRST PHASE OF A PROJECT TO DEVELOP AND UTILIZE ANIMATED DATABASE COURSEWARE

Meg Murray
Kennesaw State University
mcmurray@kennesaw.edu

Mario Guimaraes
Kennesaw State University
mguimara@kennesaw.edu

Abstract:

Database technologies have revolutionized information technologies and information systems and, as such, are recommended components of any undergraduate program in a computing discipline. However, there are many obstacles to teaching database concepts including prioritizing the many important database concepts, incorporating new technologies into an already full curriculum and supporting students who find the topic challenging. This paper presents a review of the development of a set of database courseware animation prototypes designed to support and strengthen the presentation of database concepts. The work was part of a proof-of-concept NSF grant. Preliminary evaluation of the prototypes suggests they enhance student learning and, consequently, future enhancements to the courseware have been proposed.

Keywords: Database, courseware, animation

Introduction

Database technologies have revolutionized information technologies and information systems. Understanding database technologies is paramount to the success of any information technology professional and database concepts are considered foundational knowledge in all computing disciplines. In fact, the 2001 Association of Computing Machinery (ACM) computing curriculum guidelines describe database concepts, and especially relational database concepts, as among the most “important technological changes” of the last decade (ACM 2001 Joint Task Force, 2001). The IS 2002 Model Curriculum for Undergraduate Degree Programs in Information Systems specifically states that IS students are expected to exhibit proficiency with database management systems. Thus, it is no surprise that undergraduate programs in the computing and information sciences require students to engage in coursework that focuses on database principles and assign IS projects that include a database component. However, given the limited time available in a typical IS curriculum, it is difficult for students to acquire the depth of knowledge needed to attain a high level of understanding and application of database concepts and technologies.

Challenges related to the inclusion of sufficient coverage of database concepts in the curriculum have existed for years. There remains little consensus on the exact breath and depth of content that should be included in an introductory undergraduate database course due to the rapid rate of change in the field and the need to include a broad spectrum of fundamental concepts (Robbert, 2002). Further, while there is an observed need to also incorporate newer technologies, little investment has been made to improve the way we teach database design and management. Research in database pedagogy that spans several years indicates that investments in database curricula are greatly outnumbered by investments in curricula for other areas of computer science (ACM 2001 Joint Task Force, 2001).
A variety of textbooks are available that support undergraduate database courses, however, most do not cover all the important database topics, lack concrete examples, or contain explanations that are highly theoretical. It has been reported that students have enormous difficulties in such topics as database design, entity-relationship modeling and understanding SQL statements. The flat file format of a textbook may not be enough to adequately introduce database concepts. Instead, students need to be provided with a variety of learning activities. For instance, students retain only 15% of material presented in lecture and readings but by varying the teaching methods to include interactive courseware, the potential to increase knowledge retention rises by 50% (Jih and Reeves, 1992). Based on this, a proof-of-concept grant was explored and subsequently funded. The goal of this proof-of-concept was to design and develop courseware incorporating the use of animations to deepen and enrich standard presentations of database concepts and to complement database teachings as found in the most popular texts. It is believed that animations enhance the student’s learning experience by providing an alternative presentation of abstract concepts. While animation has been utilized to communicate other difficult concepts within computing technology curricula, with the exception of illustrating relational algebra concepts, a review of current literature indicates that little has previously been done to illustrate database concepts.

**Animated Database Courseware Described**

The courseware that was developed for the proof-of-concept grant, referred to as Animated Database Courseware (ADbC), includes animated tutorials and exercises and sample test questions that emphasize and reinforce concepts displayed in the animations. The ADbC currently consists of 64 animation prototypes categorized into three main modules (Database Design, SQL Queries and Transactional Processing) divided into 13 sub-modules. [Courseware prototypes for this work have been made freely available and may be found at http://coffee.kennesaw.edu](http://coffee.kennesaw.edu)

Within the Database Design Module, animations have been developed to support standard modeling techniques such as Entity-relationship (ER) and Class Diagrams from the Unified Modeling Language. Additional animations have been developed to illustrate the mapping of diagrams into relational tables, and the normalization process of database design. In mapping diagrams to tables, the student is presented with a scenario and the corresponding ER diagram is animated (see Figure 1). The student is then presented with several options for converting the data model into a relational schema. When the student chooses a solution for the transformation of the ERD, s/he is shown an animation that reinforces the correct solution, or provides information about the flaws found in the incorrect solutions. That is, the pros’ and cons’ of any option selected are graphically illustrated. The normalization component is divided into tutorials and exercises. The tutorial presents the student with a non-normalized relation or table for which the student may choose from a menu of options various graphical displays of key concepts such as functional dependencies, primary key, first normal form (1NF), second normal form (2NF) and third normal form (3NF). Normalization is illustrated for 1NF, 2NF, and 3NF. In each case, functional dependencies are highlighted. In order to decompose tables in 1NF or 2NF, the student may choose to create a new table, selecting attributes that s/he wants to put into the new table. When the student finishes creating a table, the system will check for normalization violations. Specific feedback related to the violation is displayed then. When the student is finished creating all tables, the system will check each table for 3NF violations. If all the attributes from the original table are present in the new tables, then appropriate feedback is given to the student. Figure 2 presents the results of conversion to 3rd normal form.

![Module = E-R Diagram to Table](image)

---

**Figure 1:** Scenario for (1:N Relation)
Say that an Employee may be employed by one Department or No Department. Also, a Department may employ many employees.

ER Diagram:

**Figure 2:** Results of conversion to 3rd normal form.
Figure 1. Sample Animation Screen: Mapping an E-R Diagram to a Table

Figure 2. Sample Animation Screen: 3rd Normal Form

The SQL Queries Module contains sections on Animating SQL with Procedural Code, Animating SQL with Relational Algebra, an Introduction to DDL Operations (Create Table, Alter Table and Drop Table) and DML Operations (Insert, Update and Delete) and Embedding SQL Code within a Program. Animations of these exercises are composed of (1) a description window that describes the problem and the objectives of the program; (2) an SQL query window containing the SQL query that solves the problem; (3) one or more data windows containing the tables; (4) a procedure window containing the simplest form of algorithm that implements the SQL query; (5) and an output window containing the results of the query. They also contain at least two additional buttons: a “clear” button that returns the query to the initial state, and a “step” button that executes the highlighted line of the procedure window and highlights the next line every time that it is clicked. Depending on the query, there will also be a Relational Algebra window, containing the corresponding Relational Algebra solution to the exercise. These animations allow students to associate code that students may already know (procedural or relational algebra) with concepts they are learning (SQL). It also allows them to examine the execution of a query in steps. Figure 3 shows an example of the execution of a projection (select statement).
Figure 3: Sample Animation Screen: Executing a SQL Projection

The Transactional Processing Module includes animations related to Concurrency showing Serial Locking, Lost Update, Record Locking and Deadlock, as well as animations that exemplify Error Handling Mechanisms, Recovery and Triggers. Figure 5 show an example of the animation of a trigger.

Figure 4: Sample Animation Screen: Execution of a Trigger
Evaluation

Evaluation methods employed for this phase of the project included on-line evaluation forms, a comparison of test results from two introductory database classes and evaluations by faculty and students from external campuses. For the comparison of tests results two introductory database classes taught by the same instructor were used. For the first class, referred to as Class A, Database Design courseware animations were presented before quizzes and tests. No Database Design animations were presented in the second class, Class B. This process was inverted for the SQL Module – SQL animations were presented before quizzes and tests in Class B, but not in Class A. In the Database Design Module, little difference was found between the classes as they averaged approximately the same test grades (82.6 versus 82.4). For the SQL Module, however, there was a significant difference as Class B, the class that used the software, scored six points higher than Class A, the class that did not use the software (83.5 versus 77.2).

Several hundred students in eight database courses completed post-course evaluations and unanimously rated the courseware higher than other aspects of the course (instructors, textbooks, fairness and appropriate tests). On a scale from 1 to 6, where 1 is poor and 6 is outstanding, the courseware averaged 5.78 (outstanding). All other course aspects averaged between 3.1 and 4.3. Some comments included, “The E-R diagrams were great, I wish there were Normalization examples also” (which led to the inclusion of Normalization in the full proposal); “Before, I was able to answer many of these questions by memorizing the rules. With the software I really understood.” Ten database instructors from outside KSU performed a formal evaluation of the courseware, not only asking their students to evaluate the website, but also observing how their students learned from the site. Four additional faculty conducted an informal evaluation. The instructors and their students evaluated the SQL and the Converting E-R Diagrams to Tables and the Concurrency sub-modules, and filled out evaluation forms online at the Coffee website. Faculty and their students alike affirmed that the courseware enhanced student learning.

Future Work

Future work on the courseware is needed to evolve the project from a ‘proof-of-concept’ phase to a fully functional courseware system. Further work identified includes further expansion of current animation modules in the area of Database Design, including Scenarios to E-R Diagrams, SQL Misconceptions and Transactions sub-modules and inclusion of additional topics such as security, XML, and data warehousing, the development of a feedback module that records student answers and the development of ancillary materials to further support the use of the courseware in the classroom. Finally, a major task will be to convert all existing prototypes into one standard development platform and user interface. The prototypes were developed by many students, often for course credit. Consequently, the initial prototypes have been developed in 3 different platforms (Flash, VB.NET and Java). Part of the feedback collected in the evaluations shows that some of the programs created in Flash had an excessive use of animations and multimedia while some of the interfaces done in Java were too static. We plan to integrate the best of each interface, refining the prototypes and using multimedia exclusively to explain concepts and avoid its use when it does not appear to add instructional value.

Conclusion

Developing an understanding of database concepts is of paramount importance for students in any computing discipline. Database systems make up the core component of any information system. However, teaching database concepts is often challenging given curricular constraints and students often exhibit difficulties in working with abstract database concepts. One solution to this dilemma is the development of supplemental course materials. This paper provided an overview of the development of a collection of database courseware animations designed to support and strengthen the presentation of database concepts. The initial work was funded as a proof-of-concept through an NSF funded project (#0089412). Evaluation of this phase of the project was positive and suggestions for further development of the courseware were made.

References:


