DATABASE THEORY IN PRACTICE: LEARNING FROM COOPERATIVE GROUP PROJECTS*

Suzanne W. Dietrich Susan D. Urban
Department of Computer Science and Engineering
Arizona State University
BOX 875406
Tempe, AZ 85287-5406 U.S.A.
S.Dietrich@asu.edu S.Urban@asu.edu

ABSTRACT

This paper describes the use of cooperative group learning concepts in support of an undergraduate database management course that emphasizes the theoretical and practical aspects of database application development. The course project is divided into three main phases, involving requirements analysis and conceptual design, relational database mapping and prototyping, and database system implementation using Microsoft Access. The project deliverables are designed so that students not only develop a database implementation, but also evaluate their design in terms of functional dependencies, normal forms, the lossless join property, and the dependency preservation property, thus establishing the need for sound database design principles. Students are required to actively participate in each phase, with students assuming different roles in each phase to allow them to experience different leadership responsibilities. As part of the grading process, students evaluate their own performance as well as the performance of others in the group. This paper describes our experience with the structure and administration of cooperative groups and provides a discussion of the lessons we have learned, including initial observations of the effectiveness of the approach.

INTRODUCTION

Database education has matured to the point where there is a strong base for teaching database theory and, at the same time, a strong practical side to the use of database management systems [6]. Database theory is important, not only because of the foundation it provides for research, but also because it is a prerequisite for effective use of database management systems [1]. Database practice is important for preparing students to address the needs of real-world applications. There is, therefore, a wealth of both theoretical and practical material that must be covered within a typical undergraduate course on database management systems.

Another concern with respect to database education is the overall approach to teaching database concepts, such as the use of cooperative learning [4] in the college classroom. In cooperative learning, in contrast to *individual* or *competitive* learning, students work in small groups, helping each other learn the assigned material. Cooperative learning has advantages that go beyond simply learning the course material. In particular, cooperative learning often emphasizes oral and written communication and provides students the opportunity to improve their leadership capabilities. These issues are an orthogonal aspect of the educational process that many employers view just as important as technical capabilities.

Prior to the 1994-95 academic year, the undergraduate database management class in the Department of Computer Science and Engineering at Arizona State University, CSE 412, emphasized theory rather than practice due to: 1) the dilemma of what to teach from the myriad of topics available for a 15 week database management class, and 2) a lack of database computing resources. At that time, we only used cooperative learning in the database classroom through informal groups working on study problems. In the 1994-95 academic year, we modified the course structure of CSE 412 to provide a better balance between the theory of database management systems and the practical application of that theory. These changes have been supported by a recent award from the National Science Foundation under the Leadership in Laboratory Development Projects component of the Education and Human Resources Program (Grant No. DUE-9451489). In particular, the curriculum of CSE 412 covers: an introduction to database systems; entity-relationship conceptual modeling; database design; and relational databases. Specifically, the relational model emphasis includes: mapping from an ER diagram to a relational schema; relational query languages and query optimization; relational database

*This research was supported by NSF grant DUE-9451489.

SIGCSE '96 2/96 Philadelphia, PA USA
©1996 ACM 0-89791-757-X/96/0002....$3.50

design theory; transactions, recovery and concurrency control; security and integrity. The curriculum changed by removing coverage of legacy databases, such as network and hierarchical databases, and introducing cooperative group projects that focus on the practical aspects of databases using relational database technology.

This paper describes our experience with the use of cooperative group projects in CSE 412. The course project is divided into three main phases, involving requirements analysis and conceptual design, relational database mapping and prototyping, and database system implementation using Microsoft Access. The project deliverables are designed so that students not only develop a database implementation, but also evaluate their design from a theoretical point of view, thus establishing the need for sound database design principles. Students within each group are required to actively participate in each phase, with students assuming different roles over the course of the project to allow them to experience different leadership responsibilities. As part of the grading process, students must evaluate their own performance as well as the performance of others in the group. The group projects have been well-received by the students, most of whom appreciate the opportunity to relate the theoretical and practical aspects of databases. The students have also been quite receptive to the overall structure and administration of the projects, much of which has been refined several times over the last year.

After describing related work on database group projects, we then elaborate on our approach to the use of cooperative group projects. The phases of the project and their deliverables are described, followed by a discussion of the structure and administrative details of groups. The paper concludes with a summary of the lessons we have learned on the use of cooperative group projects.

RELATED WORK

Other recent investigations into teaching an undergraduate database management class include the work in [1, 5, 7, 8]. Apenyo [1] describes a two semester database management course, where the first course focuses on theory and the second course focuses on practice with students working in teams to develop an actual database design and implementation, rather than the one semester course that our work is targeting. Leeper [5] also describes a team design project for a database management course but focuses on the integration of individual subportions of the project. Other approaches to database team projects are described by Saiedian and Farhat [8] and by Pigford [7]. Saiedian and Farhat describe an approach where each team implements a database application in six different phases with documentation due after each phase. Pigford [7] describes an approach to database design teams that is close in content to the team approach used in our curriculum. Pigford's work, however, does not describe the organization of the team itself. In fact, most of the above projects have not focused on the different roles that exist in team projects and the way in which students can have the opportunity to assume different roles. The projects that we examined also did not mention team evaluation of their own performance and participation as part of the assessment process.

PROJECT CONTENT

The semester-long group project is assigned the second week of the semester and is due the last week of classes. Each group is responsible for identifying the specific application to be developed and the expert in the application area to provide input in the design and development of the database. The project is divided into three phases:

1. Requirements analysis and conceptual design,
2. Relational database mapping and prototyping,
3. Database system implementation.

Each phase has specific assessment and technical deliverables. The assessment deliverables, described in the next section, include group status reports, team assessment and confidential evaluations. The remainder of this section details the technical deliverables of each phase.

Phase 1 is the requirements analysis and conceptual design phase, currently allocated 4 weeks. Each group identifies the specific application to be developed, requiring an expert user in the application area to provide input to the design. The technical deliverables due at the end of this phase include: a description of the requirements; an Entity-Relationship (ER) diagram with structural constraints specified; a list of constraints that are not captured on the ER diagram; and a summary of processing needs, categorized with respect to expected forms, reports and queries. Two weeks into Phase 1, the group must hand in both the requirements description and the ER diagram for review. The instructor provides feedback on these intermediate technical deliverables by the next class meeting. This feedback is based on the consistency of the requirements description and the ER diagram, as well as the scope of the project. Also, at the end of this phase, each group must give a brief class presentation on the enterprise that they have chosen for their project. This presentation is given by a group-designated speaker, who is typically the group liaison to the expert user.

Phase 2 is the relational database mapping and prototype phase, which is currently allocated 5 weeks. After receiving detailed comments on the conceptual design in Phase 1, the group may need to refine the ER diagram before beginning the definition and prototype of the application. The intermediate technical deliverables include: a refined ER diagram; a relational schema for the enterprise that indicates the attributes and keys of each relation; a list of functional dependencies holding on the enterprise; and a discussion of their relational design with respect to normal form, lossless join and dependency preservation properties. This gives the students the opportunity to apply the design theory learned in class to their realistic application. The technical deliverables at the end of the phase include updates of the intermediate technical deliverables and a prototype of the database implemented in Microsoft Access. This prototype includes the definition of the schema, including referential integrity as defined via the relationships window in Access, and a prototype of the forms and reports listed. Each group member must be responsible for the prototype (and later implementation) of at least one form, one report and one query from the processing needs identified in Phase 1. Only the look and feel of the forms and reports are required for this phase.

Phase 3 is the database system implementation phase, which is allocated 4 weeks. There are no intermediate technical deliverables for this phase (although a detailed group status report is required as an assessment intermediate deliverable). This phase is dedicated to the correctness of the implementation of the database system, including robust test data. The technical deliverables due at the end of the phase include a user's guide for the application and the Access implementation. The user's guide provides: an overview of the system; a description of the forms, reports and queries; and an appendix consisting of technical support documentation, such as the ER diagram, the relational design, and a description of the implementation of explicit constraints. The Access implementation is demonstrated by the group, typically during the week of final exams.

GROUP ADMINISTRATION

Each group consists of a team of 4 or 5 students. We hand out a questionaire during the first week of class to gather information on the students to help form groups. Initially we are using prior database experience and geographical area within the Phoenix metropolitan area to form teams such that each group has at least one member with database experience and most group members live in the same region (if possible, to facilitate group meetings).

The assignment of complementary and interconnected roles to group members is an important component of cooperative learning [4]. The roles that we assign to the members at each phase include phase leader, phase recorder, phase checker and phase technical advisor(s). The phase leader is responsible for coordinating the activities of the phase, establishing intermediate deadlines and ensuring the on-time completion of the deliverable. The phase recorder establishes an outline and plan for generating the phase documentation by delegating subportions of documentation to other group members, and ensures the correctness and quality of the final documentation. The phase checker is responsible for gathering and reviewing the assessment for each group member, including the checker, and reporting the results of the assessment. The phase technical advisor serves as technical support for the group.

The roles are assigned primarily at random during phase 1 except that the leader is the group member that has had some prior database experience. The roles for the other phases are determined so that students are assigned different roles during different phases (for the most part). The assignment of roles for both 4 and 5 member groups is shown in Figure 1, where L=Leader, R=Recorder, C=Checker and T=Technical Advisor. (Note that 5 member groups have two technical advisors.) This assignment results in the phase 1 leader having previous database experience being assigned as phase 3 technical advisor for the implementation. Other assignments are possible. We have allowed groups to petition for a change of roles in phases 2 and 3, provided that each group member is assigned a different role.

Phases	*4 member group*				*5 member group*				
1	L	R	C	T	L	R	C	T	T
2	C	T	L	R	C	T	T	L	R
3	T	C	R	L	T	L	R	T	C

Figure 1: Role Assignment

Groups are formed the second week of class. Each group member signs a statement that acknowledges the importance of their membership in the group and the penalty of withdrawing after groups are formed (a failing grade within the course). Students also acknowledge that failure to contribute to the project results in a grade of zero. Each team *must* turn in the appropriate documentation at each assigned due date, even if it is incomplete. In particular, each phase has an intermediate and final phase deliverable. The intermediate phase deliverable is not graded but is used to provide feedback to the group for improving the final phase deliverable. Satisfactory performance in each phase is required before going on to the next phase. Teams that do not meet the minimum requirements for satisfactory work are required to revise their work before moving on to the next phase. The group project is worth 35% of the total course grade: phase 1 is 8%, phase 2 is 12% and phase 3 is 15%. An individual's grade on the group project can be adjusted (as described below).

At the end of each phase, each group member *must* submit a confidential phase evaluation that evaluates each group member's participation and contribution on the deliverable. This evaluation is both a self evaluation and a peer evaluation. When there is a consensus on the lack of the group member's contribution, points are deducted from that member's grade. Generally, the self evaluations are consistent with the peer evaluations, confirming any lack of participation as well as excellent contributions.

At the phase intermediate and final deliverables, a group status report is required. The status report provides the dates and attendance at group meetings during the phase, an overview of the progress of the project and a detailed list of expected and completed contributions for each group member. The group status report is primarily the responsibility of the phase leader with assistance by the phase recorder. However, each group member must sign the report, indicating that they have reviewed the report for correctness. The group status report gives a good indication of the division of labor among group members. At the intermediate deliverable, the report gives the instructor the opportunity to provide feedback on the delegation of responsibilities.

Also required at the end of a phase is a phase assessment that evaluates how well team members understand the relationship between the theoretical components of the course material and the practical aspects of the course project. The assessment is performed by the phase checker to create and encourage an environment in which peers help peers. The format for team assessment is provided by the instructor and may be a combination of a written quiz, oral discussion questions, or a self-assessment checklist. In all cases, the phase checker must evaluate the responses of each team member and provide feedback. The phase checker must also provide a detailed report of the assessment process as part of the phase deliverable.

LESSONS LEARNED

CSE 412 has been taught for two semesters now with the incorporation of group projects. This section discusses some of the lessons that we learned from this process, including intermediate deliverables, group status reports, presentations, assessment and electronic dissemination.

Intermediate deliverables were not built-in to the first offering. During the conceptual design phase, each group was requested to meet with the instructor during office hours to discuss their design. We formalized this checkpoint in the second course offering and introduced an intermediate deliverable for phase 2. The intermediate deliverables provide an opportunity for the instructor to give useful feedback to the students before they complete the current phase. We have noticed significant improvement in the final phase deliverables since intermediate deliverables were added to each phase.

The group status reports in the intermediate and final phase deliverables were introduced in the second offering of the revised course. In the first offering, we required *weekly* reports both from an individual and from the group by electronic mail (email). This was a nightmare! Students would forget to send an email and when they did (more often than not), the reports were not detailed enough to know which group member was responsible for what. The detailed group status reports consisting of the required format worked well in the second course offering.

We have tried two forms of presentations in the first year. In the first offering, we dedicated a class period at the end of each phase for detailed presentations (20-25 minutes) by some (2-3) groups chosen at random and by members of those groups chosen at random. By the end of the semester, the class was aware of the various projects. However, the feedback that we received that first semester indicated the students wanted to know this information at the end of phase 1. In the second offering, we also dedicated a class period at the end of each phase for brief presentations (10 minutes), which we called phase summaries, by all groups. Again, the group member to present the phase summary was chosen randomly. The feedback we received indicated that students appreciated the presentations at the end of phase 1 but found the presentations at the end of phases 2 and 3 to be less useful. Students also wanted the ability to designate the group member to give the presentation. The next offering of the class will only have brief presentations at the end of phase 1 by a group-designated speaker. This will free up one week of class time, which we plan to use in providing an in-class Microsoft Access laboratory assignment. Currently, the students are responsible for learning Access on their own outside of class. Overall, this works quite well. The in-class assignments, however, will allow us to acquaint students with some of the more difficult aspects of Access that are useful for the project.

In the first year of the course, the *phase checker* was responsible for checking the relationship of the technical material learned in class to the practice of that theory in their specific group project. In the first offering, the phase checkers were given complete flexibility for the design and implementation of their assessment questions. Due to the creativity (or lack thereof) of phase assessments in the first offering, the second offering of the class required an assessment proposal as part of the intermediate deliverable. The feedback that we received indicates that student's appreciate the goals of the assessment but feel that the questions should be supplied by the instructor rather than the phase checker. The phase checker still has the responsibility to collect and discuss the result of the assessment with each group member. The next offering of the course will have instructor-supplied assessment that has been motivated by the (best of the) assessment questions observed in the first year.

We plan to take advantage of this change in assessment to motivate students to use the curriculum materials that have been developed for the class. A full implementation of a company enterprise from the course textbook [3] has been implemented by an undergraduate assistant to give the students an Access example of an enterprise with which they are thoroughly familiar. Unfortunately, most students did *not* look at this example. To facilitate the students' exploration of this example implementation, we plan to have the instructor-supplied assessment consist of two parts: one that relates theory to their specific application and another that relates theory to the company enterprise and its implementation in Access. We feel that this new assessment in conjunction with the proposed in-class (one week) laboratory on Access will provide the students with a better foundation for the prototyping and implementation phases of the project.

Another lesson that we learned includes the electronic dissemination of the course material. In the first year, the sample phase deliverables were only available in hard-copy form through reserve in the library. (This was partly due to time constraints on the development of these materials that prohibited some electronic form of dissemination.) Unfortunately, we had to deal with out-of-order or missing pages. By establishing a homepage on the world wide web at the following address:

`http://www.eas.asu.edu/~cse412`

for our database management class CSE 412, we not only provide the students at ASU with the curriculum material but the entire educational community also has access to the details of how we organize and administer the group projects. We have included access to all of the forms for the various phase deliverables, the implementation of the company enterprise in Microsoft Access, and the sample phase deliverables related to the company enterprise. The web page also includes information on the RDBI educational tool [2] that provides a Relational DataBase Interpreter for relational query languages: relational algebra, domain relational calculus, tuple relational calculus and SQL. This tool has been used for several years at ASU as an intricate part of the database management class. RDBI assignments are given strictly as individual student assignments, complementing the group work that the students experience as part of the course project.

An initial observation based on examining student grades from the first year of the group project to the previous academic year shows a substantial increase in the number of A, B and C letter grades earned by the students with a corresponding substantial decrease in the number of D, E and W letter grades. The number of withdrawals (W) is obviously related to the assignment of a failing grade (E) in the course once the instructor's signature is required for a withdrawal. (Students may withdraw without an instructor's signature through the third week of classes.) The few failures in the first year are due to students who failed to participate in the class and the group project, usually during the implementation phase. The effected groups must meet with the instructor to detail a reduced implementation plan based on the loss of that group member. Although the students' letter grades are higher (due to the group project), the students' individual performance on exams has decreased. This decrease appears to be due to a focus on implementing the group project rather than studying for exams. We are considering exploring cooperative learning techniques to motivate groups to study together for exams. The students' individual performance on the RDBI assignments appears unchanged. More detailed evaluation is beyond the scope of this paper but is scheduled as part of the contributions of the grant.

We have also learned that, although the group projects add a substantial effort on both students and instructor, the extra effort is definitely worthwhile. The students appreciate the opportunity to use a real database product and an overwhelming majority of the students appreciate the group experience, learning far more than database topics from their fellow group members. As an instructor, the group projects provide an additional avenue of communication with the students that gives the instructor the opportunity to know the students better.

ACKNOWLEDGEMENTS

This work was supported by NSF's DUE ILI-LLD program (DUE-9451489). We would like to thank the students who took CSE 412 in the 1994-95 academic year for their constructive feedback. We are indebted to Rene Chan for his invaluable support in developing curriculum materials and being the Microsoft Access technical advisor to the class in the first year of the grant. We also wish to express our gratitude to Patricia Baker and Michael Tjahjadi for their work on establishing the CSE 412 homepage.

REFERENCES

[1] K. Apenyo, "A Database Sequence: Theory Then Practice," *ACM SIGCSE Bulletin*, vol. 22, no. 4, December 1990.

[2] S. W. Dietrich, *An Educational Tool for Formal Relational Database Query Languages*, *Computer Science Education*, vol. 4, pp. 157-184, 1993.

[3] R. Elmasri and S. Navathe, *Fundamentals of Database Systems*, 2nd edition, Benjamin Cummings, Redwood City, 1994.

[4] D. W. Johnson, R. T. Johnson and K. A. Smith, *Active Learning: Cooperation in the College Classroom*, Interaction Book Company, 1991.

[5] R. Leeper, "A Project Course in Database," *ACM SIGCSE Bulletin*, vol. 22, no. 1, February 1990.

[6] A. Motro, *What to Teach about Databases*, Panel Discussion, *Proc. of the 1993 ACM SIGMOD Intl. Conf. on Management of Data*, pp. 420

[7] D. V. Pigford, "The Documentation and Evaluation of Team-Oriented Database Projects," *ACM SIGCSE Bulletin*, vol. 24, no. 1, March 1992.

[8] H. Saiedian and H. Farhat, "A Team-Oriented, Project-Intensive Database Course," *ACM SIGCSE Bulletin*, vol. 23, no. 1, March 1991.