<table>
<thead>
<tr>
<th>Title</th>
<th>Towards a minimalist approach to lesson planning of an introductory database course</th>
</tr>
</thead>
<tbody>
<tr>
<td>Author(s)</td>
<td>Kwan, ACM</td>
</tr>
<tr>
<td>Citation</td>
<td>The 2nd IEEE International Conference on Teaching, Assessment and Learning for Engineering (TALE 2013), Kuta, Indonesia, 26-29 August 2013. In Conference Proceedings, 2013, p. 782-785</td>
</tr>
<tr>
<td>Issued Date</td>
<td>2013</td>
</tr>
<tr>
<td>URL</td>
<td><a href="http://hdl.handle.net/10722/190182">http://hdl.handle.net/10722/190182</a></td>
</tr>
<tr>
<td>Rights</td>
<td>This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License.; ©2013 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.</td>
</tr>
</tbody>
</table>
Towards a Minimalist Approach to Lesson Planning of an Introductory Database Course

Alvin C. M. Kwan
Division of Information & Technology Studies
Faculty of Education, The University of Hong Kong
Hong Kong, China
cmkwan@hku.hk

Abstract — Research addressing learning of individual database topics is much more common than research that attempts to address broader issues in learning of database. This paper proposes a lesson planning approach for an introductory database course based on selected principles of the minimalist instructional design and the ARCS model of motivational design. The basic idea is to design action oriented lessons in a way to involve student participation as early as possible and to foster confidence of students in the learning process. The adoption of the approach results in an ordering of database topics which is significantly different from mainstream practices.

Keywords — computer science education; databases; instructional design

I. INTRODUCTION

Information management (IM) is an element of the body of knowledge in the ACM/IEEE-CS Computer Science Curricula 2013 [1] and it covers various database topics such as data modeling, relational databases, query languages and physical database design. Database management is an important course for a wide range of degree programs from computer science to library studies, and from information engineering to information management.

Much research effort on database teaching and learning has been focused on developing learning tools to enhance the learning of individual database concepts such as structured query language (SQL) [2, 3], database normalization [4], and data modeling [5]. There are some exceptions however.

In [6], Gudivada, Nandigam, and Tao argue that large sets of realistic data are needed to provide “interesting and complex database structure to demonstrate practical aspects of data modeling, SQL features, and the interplay between the database design and query performance.” The authors introduced a project approach in their database course which required each student group to select either a subset of Amazon product database or other data repositories for their projects. Progress of the group projects was in synchronism with the discussion of the corresponding project topics in the lectures. A project based approach to teach databases is also reported in [7] in which Chen, Li and Zhang advocate a constructivist approach to teaching various database concepts by guiding students to explore various database design and implementation issues in relation to the development of a database application. It is clearly that in both pieces of work, the authors taught various database topics to their students in the following order: conceptual data modeling, logical database design, database normalization, physical database design, writing SQL queries, triggers and database stored procedures, and tuning SQL queries. In fact, such an ordering to topics can be found in the content lists of many introductory books on databases [8, 9]. These books usually begin with a part on basic relational database concepts such as basic database terminology and integrity rules, followed by database modeling including entity relationship modeling, database design such as normal forms, and SQL before moving on to more advanced topics like object-oriented databases.

An Internet search for the outlines of a number of undergraduate database courses, including those for non-computer science students, reveals that most of them adopt a similar topics ordering. Such a topics ordering appears to be natural as it reflects the ordering of the related activities in the database lifecycle, starting from modeling the database requirements using the entity relationship model, then transforming it into a logical database design with the normalization principles in mind, and executing any SQL queries on the resulting database during the operation phase.

Unfortunately the “chronological” ordering of the database topics takes no consideration of the difficulties of the topics for students. Students usually find SQL easier to learn than data modeling techniques like entity relationship modeling and normalization because they can test the correctness of their database queries easily without any external help but they cannot tell whether their entity relationship diagrams are correct or whether they correctly normalize a relational database. For example, two students may produce different entity relationship diagrams for the same problem scenario but this does not necessarily mean that one of them must be wrong. Gaining help from peer is also difficult as no novices would be absolutely sure that their designs are correct. They may not even know whether their comments to their peer work are right. Another hurdle that students need to overcome is the fact that they are unlikely to be able to create an entity relationship model or a normalized database correctly at the beginning. If
students are asked to learn data modeling early in the course, their motivation of learning can be hampered when they realize their designs are still erroneous in spite of their great efforts. This paper describes an approach to lesson planning based on selected minimalist instructional principles [10] and Keller’s ARCS model of motivational design [11] in order to foster students’ confidence in the learning process by organizing the easier database topics to be covered through action oriented lessons sooner than those more difficult topics.

In the next section, a brief introduction to minimalist instructional design and the ARCS motivation model are introduced. In Section III, we detail how two mentioned models are applied to create a new topic ordering for an introductory database course. Conclusions and a discussion of future research are given in Section IV.

II. RELATED RESEARCH

A. Minimalist Theory

The minimalist theory is a framework for the design of instruction, especially training materials of a new computer application or tool for computer users. The theory exploits the prior knowledge of the adult learners and how to use errors or anomalies as learning opportunities. The key idea of the theory to instructional design is to offer learners the bare minimum of instruction needed to get them “up and running” on some meaningful authentic tasks. Thus the theory suggests that (1) all learning tasks should be meaningful and self-contained activities, (2) learners should be given realistic projects to work on as quickly as possible, (3) instruction should permit self-directed reasoning and improvising by increasing the number of active learning activities, (4) training materials and activities should provide for error recognition and recovery and, (5) there should be a close linkage between the training and actual system [12].

While the two studies described in [6, 7] also emphasize the use of realistic project, they do not provide clear guidelines to instructional design as the minimalist theory does.

According to the minimalist theory, all learning tasks should be self-contained activities. If data modeling technique like entity relationship modeling is covered before introducing normal forms and SQL, learners may find it hard to appreciate the value of creating a data model. If the learners are told that an entity relationship model can be transformed into a database design conforming to the normalization principles, they may find it even more puzzling as they know nothing about normalization at that stage.

In order to comply with the guideline that instruction should permit self-directed reasoning, teachers can design learning tasks to empower their students to exploit errors and to investigate fixes. Teachers should identify common errors and/or misunderstanding of concepts and embed them into student exercises. Hints, instead of solutions, to address those errors or misunderstanding can be offered to students. By adopting such a learning-by-doing approach, students can achieve a deep learning on the concerned learning materials.

B. ARCS Motivation Model

ARCS is the acronym of attention, relevance, confidence and satisfaction which are four key elements of Keller’s motivation model [11].

Attention refers to gaining learner’s attention to learning materials. It is important to keep stimulating learners by presenting effective stimuli at the beginning of and maintained throughout the learning process at a level that will arouse learners’ attention and curiosity. Variability in learning materials, examples that go against a learner’s past experience, and humor are some methods for grabbing the learners’ attention. For example, animation tools can be used to illustrate the working of more complicated SQL commands like GROUP BY.

Relevance helps learners to see the connectedness of the concerned learning to their personal needs and goals, and to link their prior learning experience with the given learning materials. Teachers should use concrete language and examples with which the learners are familiar. For example, it is easier for learners to understand what a row in a table, instead of tuple in a relation, is. A strategy to promote relevance is to help learners understand the value of the subject matter to them today and in future. In the context of database learning, teachers can stress the fact that almost all real life applications such as Facebook, banks and school library systems keep data in databases.

Confidence helps learners develop positive expectation towards their likelihood of success. Meaningful experiences also sustain learners’ confidence development during the learning process. If a student feels she cannot meet the objectives or that the cost (time or effort) is too high, her motivation will decrease. In order to foster confidence, learning tasks should be designed to allow for small steps of growth during the learning process. In database learning, students can be introduced with SQL before more difficult topics like data modeling.

Satisfaction reinforces accomplishment to maintain desirable learning behaviors. It comes when learners are allowed to use or to benefit from the newly acquired knowledge and to receive feedback in a manner that leads to positive attitudes towards the learning task. Positive feedback from teachers can help reinforce students’ satisfaction which in turn motivates the students to learn more. It is however important not to patronize learners by over-rewarding easy tasks.

III. PROPOSED LESSON PLANNING

In this section, we discuss how we organized lessons for an introductory database course designed for an information management undergraduate program at The University of Hong Kong. Most students enrolled in the program have a sub-degree qualification in business management or library studies. Like the rest of the class, they have little knowledge in computing before taking the database course.

The 24-hour course aims at offering students with basic but practical knowledge and skills in database management, covering the following database topics:
Aims of database management (e.g. files and databases)
- Relational database concepts (e.g. primary and foreign keys)
- Conceptual data modeling
- Logical data modeling
- Database normalization
- Physical database design
- Basic SQL
- Procedural SQL
- Building database applications

A. Prior Knowledge Requirement

We first looked into the topic dependency issue. Topics like procedural SQL and building database applications can only be covered after basic SQL is introduced. Similarly, logical data modeling and normalization should be taught before physical data modeling. The dependency relationships between topics, indicated by arrows between topics, are shown in Figure 1. A dotted arrow signifies a possible dependency, e.g. a database application may or may not need any procedural SQL support.

As shown in the dependency diagram, we should introduce aims of database management and relational database concepts at the beginning of the course. As to the next topic, there is a choice among basic SQL, conceptual data modeling, and database normalization.

B. Learners’ Motivation Consideration

According to the ARCS motivation model, lessons should be organized into a way to help learners develop positive expectation towards their likelihood of success. Among basic SQL, conceptual data modeling, and database normalization, we choose basic SQL as the next teaching topic because of two reasons. Compared to the other two topics, learning SQL is easier. Students can check their understanding of SQL statements by executing them on a database management system. They can learn and revise their knowledge in SQL using interactive online SQL tutorials like http://sqlzoo.net and http://www.w3schools.com/sql/. Such interactive tutorials offer students a chance to test the correctness of different answers to a problem that in turn promote self-directed active learning which is in line with the minimalist theory.

For most learners, database normalization is easier to handle than data modeling. There are well defined definitions on normal forms. By referencing those definitions, students can check the correctness of their database table designs by themselves or through peer review supported by walkthrough and think aloud techniques.

While peer review may also help identify errors in a data model, the chance of having all the reviewing learners coming to the same conclusion is slim because the learners would likely generate data models which are so different that it would be hard for the learners to determine which designs are correct.

Thus an initial ordering of the database topics is as follows:
1. Aims of database management
2. Relational database concepts
3. Basic SQL
4. Database normalization
5. Conceptual data modeling
6. Logical data modeling
7. Physical database design
8. Procedural SQL
9. Building database applications

C. Errors or Anomalies as Learning Opportunities

One way to promote self-directed learning is to direct learners to reveal mistakes and anomalies and encourage them to find solutions to those problems. For example, we can design near-solutions to database queries in order to reveal common mistakes such as using a group function in the GROUP BY clause.

Anomalies and their solutions can trigger changes in lesson plans. For example, database normalization is introduced to deal with data anomalies in relational databases. A normalized database keeps data in multiple tables. However SQL has commands that handle single and multiple tables. It is odd to introduce SQL commands for manipulating multiple tables before knowing why there are needs to store data in multiple tables. A better way to introduce the topics is to start with SQL commands that manipulate single tables. Before introducing database normalization, students can be asked to run SQL queries on a non-normalized table and guided by teachers to identify representational inadequacy due to the data anomalies involved. Teachers can also design exercises for students to reveal that while database normalization can address representational inadequacy, it does not ensure data integrity across multiple tables when such tables are updated. This
helps explain why foreign keys and referential integrity are important. After finishing database normalization, teachers can introduce SQL commands that handle multiple tables.

The ordering of the database topics for the course is revised with changes italicized as follows:

1. Aims of database management
2. Basic relational database concepts (e.g. tables and primary key)
3. Basic SQL for manipulating single table
4. Database normalization
5. More relational database concepts – foreign keys and referential integrity
6. Basic SQL for manipulating multiple tables
7. Conceptual data modeling
8. Logical data modeling
9. Physical database design
10. Procedural SQL
11. Building database applications

IV. CONCLUSIONS AND FUTURE RESEARCH

We have proposed a lesson planning approach based on Keller’s ARCS motivation model and Carroll’s minimalist theory. The approach has been used to develop the basic course structure of an introductory database course for an information management undergraduate program. This paper has been focused on how the approach affects the ordering of topics in the course. The approach in fact also offers hints to development of teaching ideas. For example, teachers should look for students’ common mistakes and misunderstanding and incorporate them into the learning materials so as to empower students to exploit those common mistakes and misunderstanding as learning opportunities.

We are currently revising the learning materials for the mentioned database course to make them more coherent with a course project that students are asked to complete. The major hurdle for achieving coherence between the teaching component and the project component is that they do not synchronize very well as the former covers data modeling significantly later than what the latter requires.

In the future, we plan to perform an evaluative study of the revised database course regarding whether it can facilitate a better learning experience for students, and whether it can improve the students’ learning. The usual evaluation approach based on an experimental group and a control group is unlikely to be appropriate for this study for two reasons. First, due to a lack of space resources, all students taking the revised database course will be taught in mass lectures. Second, we feel that it is not ethical to give students different treatments in a course in which their assessment grades will contribute to the students’ classes of degree. Further investigation for an appropriate evaluation approach is needed.

It is worth mentioning that our proposed lesson planning approach is context-free as its guiding principles are generic. The approach is however more suitable for skills based training and courses with plentiful hands-on opportunities. Many computer courses like web development and computer programming fall into such a category. The proposed approach is expected to be less useful for courses that involve a diverse range of loosely related topics, e.g. topics in theory of computing.

ACKNOWLEDGMENTS

Thanks are given to Dr W W Ki who has introduced me the minimalist theory and the anonymous reviewers who have offered constructive comments on an earlier version of this paper.

REFERENCES