ONLINE LEARNING AND FUN WITH DATABASES

Petra Leimich
p.leimich@abertay.ac.uk

Leslie Ball
l.ball@abertay.ac.uk

School of Computing and Advanced Technologies
University of Abertay Dundee
Bell Street
Dundee, DD1 1HG, UK

ABSTRACT

In this paper, we explore how online learning can support face-to-face teaching in fundamental database theory and the contributions it can make towards motivating and enhancing the student learning experience. We show how we have used WebCT for a third level database module and present student feedback to our approach. While online participation is high overall, motivation for self-learning is increased by the use of self-assessment exercises and summative assessment was also considered to be more fun online than using paper based equivalents. Evidence exists to link greater online participation of course materials to improved performance. We complement our feedback by presenting and discussing a number of software tools which help students practice important methods in database systems, including SQL. After evaluating these against known methods for improving student motivation, we suggest ideas for further development of more game-like learning tools.

Keywords

Database learning; Interactive tools; Motivation.

1. INTRODUCTION

Students everywhere find it difficult to understand a number of concepts and methods which are fundamental in database theory [5, 8, 10] and pass rates for theoretical database modules are often very low. One reason for this, in our experience, is that too many students fail to engage with the material. So how can we encourage students to spend more time with a subject that many perceive as difficult and dry? Finding ways to improve student motivation could go a long way towards better performance. Having taught database theory for several years, we have tried a number of approaches. So far, providing web-based support for face-to-face teaching appears promising, even though we provide little more than weekly self-assessment quizzes (SAQs) in addition to copies of lecture notes etc. Concepts like data modelling, normalisation and SQL are essential in any database course and are often underpinned with others such as relational algebra and index structures. The drive is thus towards improving motivation in these particular areas.

Teaching, Learning and Assessment in Databases, Coventry 2003

2. ONLINE DATABASE MODULE SUPPORT FOR THIRD LEVEL

We have been using WebCT for several years to provide online support for module IC301A, Designing Databases. This is a core module for Computing, Computing (Applications Development), Business Computing and Information Management students at the University of Abertay Dundee and has around 130 students each year. For almost all students, this is the first time they are exposed systematically to theoretical database concepts. The module runs for 12 teaching weeks and offers a 1 hour lecture and a 2 hour lab/tutorial each week. The WebCT support is available through the internet anywhere, anytime to registered students. Materials include all lecture notes, lab/tutorial exercises, additional notes for a number of topics, and weekly SAQs. One reason for selecting WebCT as the environment for online support was that it supports the easy creation of quizzes. Each SAQ focuses on one topic and consists mostly of multiple choice and a few short answer questions. Images are used extensively. Students can take each SAQ as often as they like, and will get instant feedback after submission. In addition to their score, this includes the correct answers as well as specific explanations for some common mistakes. Assessment for the module comprised two short class tests and a 1.5 hour examination. The class tests were taken online in WebCT in a supervised environment and followed similar formats to the SAQs with the addition of some questions which required paragraph answers and tutor marking. Test results were then released in WebCT.

2.1 Student feedback

Students were invited to give detailed feedback about their experience of the module and specifically

on the online support via an anonymous online survey. The completion rate was nearly 90%.

2.1.1 General feedback

Students used the online support extensively: 96% at least once a week, and 39% more than twice a week. As expected, students agreed almost unanimously that the online support helped them catch up if they missed classes, and that they found it useful overall. We also found that the online support improved student motivation – 78%, 76% and 67% respectively agreed or strongly agreed that it helped them enjoy the module more, learn more independently and spend more time on the subject than they would have without the online support.

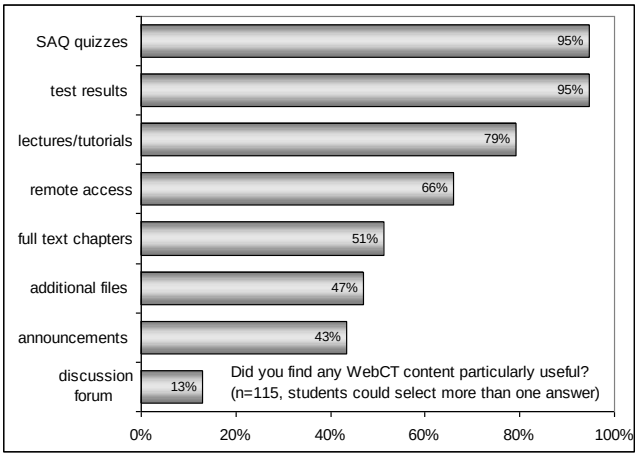

Figure1: Usefulness of WebCT content

Figure 1 shows that students found the SAQ quizzes the most useful (95%). Despite photocopies of lecture notes and tutorial sheets being given out in class, students rated online access to them very highly. Remote, off-campus access was named by 66% of all students as particularly useful. In our analysis of the survey results, we found no statistically significant differences between direct entrants to year 3 and other students.

However, we did find that students' enjoyment of WebCT varied significantly with their course of study (Figure 2). Applications Development students enjoyed WebCT considerably less than students on other courses, while Information Management and Business Computing students showed the highest proportion in strong agreement.

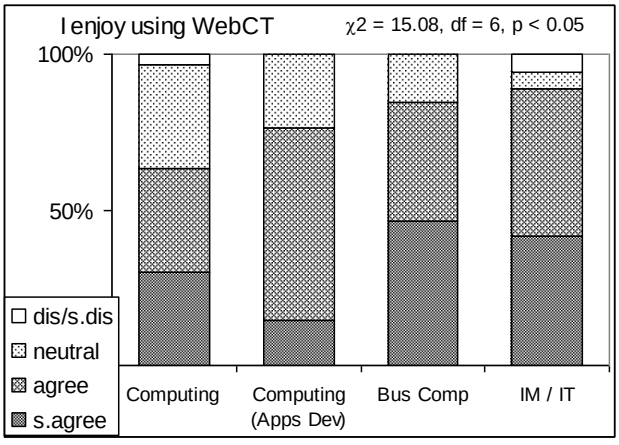

Figure 2: Student feedback by course

2.1.2 Quiz Feedback

In addition to the general questions, we asked for more detailed feedback about the online SAQs and tests. The results are summarised in Figure 3. As before, the feedback was overall positive – 92% of students said that the SAQs helped them understand the material better, while 55% strongly agreed that the online summative tests were more enjoyable than paper-based equivalents. Interestingly, of all aspects covered by questions, students were least enthusiastic about their enjoyment of the SAQs.

	Strongly agree	Agree	Neutral	Disagree	Strongly disagree
I enjoyed doing SAQ	35%	80%	97%	99%	100%
SAQ help me understand material better	42%	92%	98%	100%	100%
SAQ feedback deepens my understanding	36%	84%	99%	100%	100%
SAQ were good preparation for tests	45%	83%	96%	100%	100%
Online tests were more enjoyable than paper	55%	86%	96%	97%	100%

Figure 3: SAQ feedback (cumulative percent)

2.1.3 Discussion

Overall, student feedback has been very positive. This is also reflected in written comments. One unexpected side effect of using online quizzes and an online feedback survey was that more than half of the respondents added detailed free text comments. For paper surveys this figure is usually very low. Several students suggested that similar WebCT support should be provided for all their modules; others commented that the SAQs should be more taxing, that they found it less daunting to check their progress or do the formal assessments online, and that the SAQs helped pinpoint weak areas. See [6] for a more detailed evaluation.

2.2 Performance analysis

Given the positive nature of student feedback, we hoped that the added support and enjoyment might contribute to an improvement in student performance. This is difficult to evaluate, as we make changes to the overall content and delivery of the module every year and the student cohorts vary. We have not run controlled experiments, as we wanted all students to be able to benefit from all available support. Comparing pass rates year on

year, we can report that this improved from 72% in 2001-02 and previous years to 81% in 2002-03. However, this cannot be due to online support alone, as this was available in almost identical form in the previous year. What did change was the integration into the module as a whole. Rather than just suggesting an SAQ during the lecture, along with reading, we required students to attempt a specific SAQ quiz every week in preparation for the tutorial. While this was not assessed formally, students were not allowed to attend the tutorials unless they had attempted the quiz. With most students, this had the desired effect – the participation rate in SAQs increased dramatically compared to the previous year. However, a substantial minority of students simply decided not to attend.

As expected, there was a significant correlation

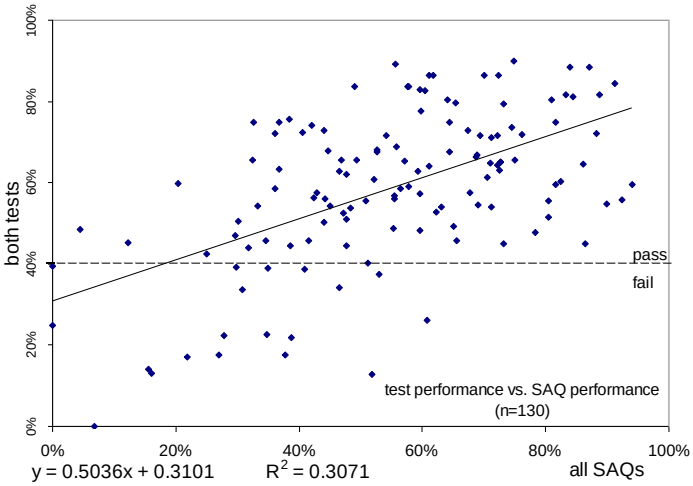

Figure 4: Test performance vs SAQ performance

between performance in the SAQ attempts and the summative online tests (Figure 4). Thus, the higher the total mark achieved in SAQs, the better the expected assessment result.

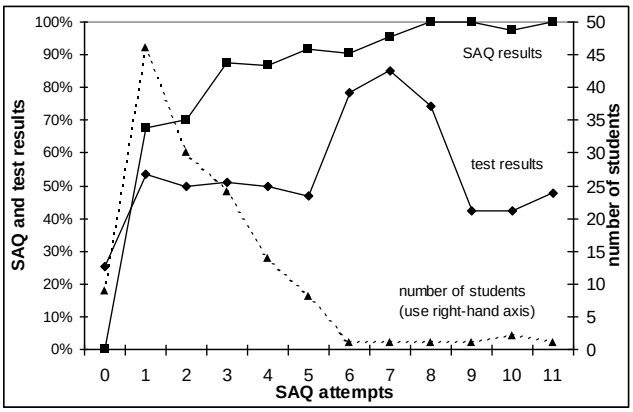

Figure 5: SAQ and test results vs. SAQ attempts

Figure 5 shows that many students tried the SAQs more than once – some students did so up to 10 or 11 times. Those who made no SAQ attempt performed poorly in the test. Interestingly, however, student performance did not improve systematically with repeated SAQ attempts. This may be due to the static nature of the SAQ quizzes – each attempt would repeat the same questions. It may be important to provide more different questions in order to ensure that student application skills really improve rather than just learning specific answers by rote. Also, some students suggested that the SAQ questions were not difficult enough. We hope to add more interactive content in the future, and discuss in the following sections which form this might take.

3. INTERACTIVE DATABASE TEACHING TOOLS

There are a number of interactive database teaching tools in the areas of SQL, ER modelling and normalisation. What can these tools offer?

3.1 Tools for SQL

esql [5] is a complete DBMS coded in C, with a conventional graphical user interface. It is designed to enhance student learning by providing a 'Help mode', in which students can go through the execution of any SELECT query step-by-step. esql then shows intermediate results and an explanation for each step, as well as highlighting the relevant part of the SQL statement. This method helps students gain an intuitive understanding for how queries work. Currently, SELECT queries are implemented including joins and aggregation, but not nested queries. esql also implements some DDL, which allows students to create their own database schemas.

WinRDBI [2, 11] is a teaching tool not only for SQL, but also for the formal query languages of Relational Algebra and Relational Calculus. It can be freely downloaded [11]. The heart of WinRDBI is a relational database interpreter, written in Prolog, supplemented with a graphical user interface. As with esql, students can create their own database schema and run an unlimited number of different queries.

SQL-Tutor [8, 9] is available as shareware [4]. Access to a web version, included in an environment called the Database Place, is distributed free with some textbooks [1]. With SQL-Tutor, students are given query descriptions in natural language and write SQL solutions. The solutions are then evaluated by comparing them with stored solutions using a rule-based system. SQL-Tutor provides feedback to the student, who can select from different levels. These include a detailed listing of errors (one at a time), the correct answer, or hints for the next line. As SQL-Tutor uses a fixed set of exercises and stored solutions, the student queries can be evaluated not only for syntax but also for semantic correctness. For example, feedback could flag up if a Cartesian product is used inadvertently

rather than a join, by missing out a join criterion in the WHERE clause. This is a feature that none of the other tools can offer.

SQL-Tutor is an intelligent tutoring system (ITS), which uses a constraint based modelling approach to create a student model. Thus, while the student is concentrating on solving each problem, the student model learns about the student. This allows subsequent problems to be selected from the archive according to individual student progress. For example, the system might select examples that provide further practice with clauses which the student has found difficult. If they wish, students are free to select the next problem themselves. They can also check their overall progress with the student model.

3.2 Other tools

KERMIT [10] is an ITS like SQL-Tutor, but for entity-relationship (ER) modelling. It provides a problem-solving environment, where the tool gives a verbal description of a scenario which students then translate into an ER model. KERMIT facilitates learning by providing guidance at any stage if requested by the student, and by evaluating student answers and giving detailed feedback on any errors made. Like SQL-Tutor, the tool adapts to each student and can select new problems specifically to suit the student.

NORMIT [4] completes the family of intelligent tutoring systems for database topics developed at Canterbury, New Zealand. Like SQL-Tutor, it is available at the Database Place [1]. NORMIT provides a large number of examples where students are given the functional dependencies of a relation and then work out candidate keys, primary key and simplified functional dependencies, finally determining which normal form the relation is in.

3.3 Evaluation

How useful are the tools introduced above in supporting student learning?

SQL-Tutor has been evaluated extensively [9]. This shows that SQL-Tutor is perceived positively by students, and has improved their performance substantially. Almost all students would recommend the tool to others, and in the most recent study, about 40% of students stated that they enjoyed learning with SQL-Tutor very much. This high figure reflected changes made to the system after earlier student feedback. For example, scenarios different from those used in the lectures and tutorials were perceived as more challenging and more enjoyable by the students. The study also showed that even a single session with SQL-Tutor led to a significant improvement in student performance.

An evaluation of Kermit [10] also showed a considerable improvement in student performance. Student feedback was positive; most students enjoyed using the tool. We are unaware of any formal evaluative studies of esql and WinRDBI, and an evaluation of Normit is not yet available [4].

3.4 Discussion

As shown by the evaluations, SQL-Tutor and Kermit are undoubtedly valuable tools which enhance traditional classroom teaching methods. It is likely that an evaluation of any of the other tools would also show positive results. Are there any drawbacks? Each of the systems is a stand-alone tool. As students will need to familiarise themselves with each environment, it is questionable whether all of these tools would really be suitable for short-scale use within a general database module.

Most of the tools found were for SQL. This suggests that this may be the topic area with the greatest need. Also, the closed and algorithmic nature of SQL may make it the most amenable for the development of teaching tools. SQL-Tutor is the most advanced of the SQL teaching tools introduced here, as it is the only intelligent tutoring system. However, the fixed problem set is a drawback, as students are forced to work with given scenarios and exercises. This is offset by offering a number of different scenarios and enough problems to ensure that students will have a good grasp of SQL by the time they complete the set exercises. Another problem inherent in SQL-Tutor is that some creative correct solutions are not recognised as correct, if they differ substantially from the stored solution. This could be very frustrating for students.

Even though none of the tools are designed explicitly to enhance student enjoyment or fun, students generally enjoyed using SQL-Tutor, Kermit and our WebCT support with its self-assessment quizzes, and staff observed an improvement in motivation. One student confessed: “When I first started the module I must admit I found the whole subject dull. However to my surprise it developed into a very interesting module.” Unfortunately, student attendance indicates that a substantial minority of students disagreed. This is reflected in poor performance for this subgroup. So how can we motivate more students? Is it possible to design effective tools specifically to enhance fun and motivation, especially for the weaker students? Can we learn from games?

4. MOTIVATING STUDENTS

Much has been published about how to motivate students in tertiary education. As [7] points out, all students are different and enter University with a history of different learning experiences. Therefore, there is no magic solution that could highly motivate all students. [3] and [7] list a number of motivating factors identified by students themselves, which include instructors’ enthusiasm, rapport between instructors and students, active involvement of

students and organisation of the course. Materials should be relevant, use concrete and understandable examples, be of an appropriate difficulty level, and provide variety. Based on this, [3] makes a large number of practical suggestions, recommending that the difficulty of the material should increase over time, in order to give students early opportunities to succeed; early, immediate and frequent feedback is essential. Grades should be de-emphasised, for example by allowing students to improve them after feedback. Lecturers should hold high but realistic expectations of their students, and these should be made clear. Another recommendation important in this context is to use a variety of methods and make students active participants.

These criteria explain the success of the tools discussed earlier and of our online support. As all were used in addition to traditional methods, all would have increased the variety, given the opportunity to succeed, clarified expectations and enhanced feedback. Normit scores quite low, as its exercises are very theoretical. Only relations and attributes named A, B, etc. are included. Also, the functional dependencies are given in the problem, while they would have to be determined from data or attribute names in the real world. By contrast, SQL-Tutor and Kermit in particular rate very highly, as they also adjust the difficulty level to suit each student individually. While the WebCT support we offer does not achieve this, it organises the whole module very clearly.

5. CONCLUSIONS AND FUTURE IDEAS

Expanding the use of WebCT beyond simple delivery of lectures by introducing self-assessment and summative assessment methods led to greater module participation and improved performance. Student feedback has endorsed the view that online materials can enhance the learning experience. This feedback along with an investigation into current teaching tools for database methods has given rise to new opportunities in the area of game-based learning.

Our analysis clearly indicates some ideas for improving our online support. For example, we could expand the quizzes so that students repeating a quiz get different questions each time. Furthermore, these could increase in difficulty. WebCT makes it easy to conditionally release quizzes based on past performance. The tools discussed here were designed mostly to give students more practice and experience with important practical aspects of databases. However, we suggest that fairly simple animations could be created which could aid students in understanding ideas in the first place. Java applets could achieve this and be extended to be interactive. This could be especially useful for visualising the algorithms behind B-trees etc. Drag-and-drop tools could provide appealing interfaces for practicing relational algebra, SQL, modelling and building small B-trees by hand. Instead of giving marks, students could collect new additions to their “toolkit” for successful completion of a task, which would then let them move on to harder problems. This would create a more game-like feel. For example, students might start an SQL game just with SELECT, FROM and WHERE keywords and “earn” additions such as COUNT and GROUP BY. The game-like feel would also be enhanced by finding a visually appealing method for showing students their progress – for example, (virtually) climbing a mountain or negotiating obstacles in a labyrinth.

6. REFERENCES

[1] Database Place. http://www.aw.com/databaseplace/.

[2] Dietrich, S. W., et al. 1997. WinRDBI: A windows-based relational database educational tool. SIGCSE'97. 126-130.

[3] Gross Davis, B. 1993. Tools for teaching. Jossey-Bass. Extract available at http://www.hcc.hawaii.edu/intranet/committees/FacDevCom/guidebk/teachtip/motiv.htm

[4] ICTG. 2002. Intelligent computer tutoring group. http://www.cosc.canterbury.ac.nz/~tanja/projects.html.

[5] Kearns, R., et al. 1997. A teaching system for SQL. Australasian Computer Science Education ACSE'97: ACM Press. 224-231.

[6] Leimich, P. 2003. Supporting on-campus courses with WebCT. 2nd WebCT European User Conference. http://www.webct.com/europe2003/viewpage?name=europe2003_programs_14_january

[7] Luce, R. W. Motivating the unmotivated. http://www.hcc.hawaii.edu/intranet/committees/FacDevCom/guidbk/teachtip/unmotiva.htm.

[8] Mitrovic, A. 1998. Learning SQL with a computerized tutor. SIGCSE'98. 307-311.

[9] Mitrovic, A., et al. 2002. Using evaluation to shape ITS design: Results and experiences with SQL-tutor. User Modelling and User-Adapted Interaction. 12, 243-279.

[10] Suraweera, P. 2001. An intelligent teaching system for database modelling. www.cosc.canterbury.ac.nz/research/reports/MastTheses/2001/mast0102.pdf

[11] WinRDBI. http://www.eas.asu.edu/~winrdbi/.