

Automated Tutoring for a Database Skills Training Environment

Claire Kenny
School of Computing
Dublin City University
Dublin, Ireland
++353 1 7005616
ckenny@computing.dcu.ie

Claus Pahl
School of Computing
Dublin City University
Dublin, Ireland
++353 1 7005620
cpahl@computing.dcu.ie

ABSTRACT

Universities are increasingly offering courses online. Feedback, assessment, and guidance are important features of this online courseware. Together, in the absence of a human tutor, they aid the student in the learning process. We present a programming training environment for a database course. It aims to offer a substitute for classroom based learning by providing synchronous automated feedback to the student, along with guidance based on a personalized assessment. The automated tutoring system should promote procedural knowledge acquisition and skills training. An automated tutoring feature is an integral part of this tutoring system.

Categories and Subject Descriptors

K.3.1 [Computers and Education]: Computer Uses in Education – *Computer-assisted instruction (CAI)*.

General Terms

Management, Design, Human Factors, Languages.

Keywords

Structured Query Language, tool-mediated independent learning, skills training.

1. INTRODUCTION

The School of Computing at Dublin City University offers a second year undergraduate module, an introduction to databases, which is presented online. Courseware is comprised of lectures in textual, pictorial, and audio format, along with animated tutorials. Also included are interactive lab sessions, one of which tutors students in SQL (structured query language), a database definition and manipulation language.

SQL is a declarative database programming language, comprised of approximately thirty English commands, such as select, insert, and update. It is the most widely used language for relational

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SIGCSE '05, February 23–27, 2005, St. Louis, Missouri, USA.

Copyright 2005 ACM 1-58113-997-7/05/0002...\$5.00.

database data definition and data manipulation. For this reason, it is commonly included in introductory database courses.

Our automated tutoring system is concerned with the SQL select statement, for a variety of reasons. This statement is used to retrieve information from a database, and can be viewed as being the most fundamental of the SQL statements. It can be simple, but also has the capacity to become quite complex, thus lending itself to being taught to students with a wide range of ability.

Learning is a complex domain. With classroom-based modules a human lecturer or tutor is generally available to personally correct answers and offer specific advice to students. This type of support is not immediately available with an online course. Disadvantages can result, such as being unable to offer personalized guidance.

Accordingly, to facilitate independent and self-reliant learning with an online environment, there needs to be a certain level of advice and guidance. Our tutoring system aims to provide this by using feedback, assessment, and personalized guidance, all of which are available at all times, to support the learner due to the absence of synchronous human tutoring. This is available at all times. The tutoring system focuses on training and development of skills rather than on knowledge and learning. It is similar in aspects to intelligent tutoring systems [10], and so interaction is primarily learner-content, rather than learner-learner.

Similar database tutoring systems exist. One such system is SQL-Tutor [7]. Like our system, it is intended as a practise environment and supposes that students have previously been exposed to the concepts of database management in lectures. SQL-Tutor uses constraint-based modelling to create a model of and deliver feedback to the student. Another system, Acharya [1], uses truth table processing to correct SQL select statements. It too offers feedback to the student.

While SQL tutoring systems exist [1] [7], it has been noted that many do not use an assessment of the student's previous progress throughout the tutorial in order to offer an optional personalized and adaptive guidance element. We aim to rectify this issue in our system. Additionally, students will submit statements to an actual Oracle database that can be used for coursework and projects, a feature often absent in other online SQL tutors.

This paper will outline the pedagogical issues we address in our project. We will then describe the conceptual structure of the system, including examples of student-system interaction. We will conclude with an evaluation and discussion.

2. PEDAGOGICAL ISSUES

Educational technology combines trusted and new information technologies with established and emerging pedagogical methods. We discuss a number of these to address the quality of automated tutoring systems.

2.1 Cognitive and Virtual Apprenticeship

Traditional apprenticeship is a process by which the student learns through observing and actively aiding the master as he completes a certain task. Through time, the apprentice will complete the task with the master looking on, and offering help when necessary.

Collins et al. [4] propose a form of apprenticeship called cognitive apprenticeship. This form of apprenticeship is suitable for tasks that are more knowledge based than those dealt with by traditional apprenticeship. Here, a given task is divided into parts. Students may then complete sub-parts suited to their level of ability and hence the complexity of the task is slowly increased. This process is carried out in an authentic setting in which the student is actively involved, thus having real-world relevance.

The virtual apprenticeship model [8] adapts the cognitive apprenticeship theory, by using scaffolding and activity theory, to make it suitable for the Web context. When developing our tutoring system we aimed to replicate components of the virtual apprenticeship model. The online tutor assumes the role of the master.

2.2 Learning Structure

Learning is often a broad and general term. It may refer to theory and knowledge acquisition. Alternatively, training and obtaining practical skills may be in question. Online learning is often focused on knowledge acquisition. Occasionally, the online environment acts simply as a replacement for a book – knowledge is presented, but skills training is not accommodated.

The automated tutoring system described in this paper is concerned with training and the development of skills rather than factual and theoretical knowledge. Students submit SQL statements to an actual database; hence they are actively involved in a realistic database programming environment.

An additional aim is to encourage guided discovery [2]. With this the primary emphasis is placed on the student and his ability to seek information, rather than the system overwhelming him with data. It is often a productive method of learning in one-to-one situations.

2.3 Learning Supports

In the traditional classroom-based learning environment, a student is usually supported while partaking in some form of learning activity. As he gains confidence and ability this support is removed, and thus the student can complete the activity independently. This support is called scaffolding [6], and its removal is referred to as fading.

In keeping with the virtual apprenticeship model, scaffolding and fading should be offered in the online classroom. In fact, scaffolding can be seen as crucial to an online tutoring system, as the teacher-student relationship is often absent.

Our system uses feedback as a means of learning support. This is described in more detail later in this paper.

3. SYSTEM DESCRIPTION

Important features of the automated tutoring system presented here include its architecture, the error classification scheme being used, the interface, and the feedback and guidance elements present. This section describes each of these features in detail.

3.1 Tutoring architecture

The system consists of four main components – interface, correction model, pedagogical model, and student model, as illustrated below (Figure 1) in the dataflow model. These are further outlined in the following subsections.

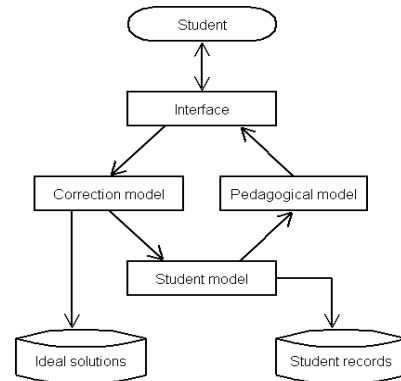


Figure 1. System architecture

3.2 Error classification

SQL is a formal language and, as with any formal language, properties of it may be *syntactical*, *semantic*, or *pragmatic*. As such, errors can also be grouped into the above three properties.

- Syntax errors are those caused by problems with the actual syntax of the language, such as misspelling a keyword.
- Semantic errors occur when the student creates a syntactically correct statement that does not reflect his intentions correctly.
- Pragmatic errors are considerably more generic and are concerned with rules and conventions within the language.

Incorrect expressions are central to the training and feedback context in question here. It is important to base error categories on the difficulties the student may encounter. These may include the correct use of basic syntax, using operations in the manner intended, or dealing with combinations of operations.

A basic implementation of the automated tutoring system analyses syntax and semantic errors. These are further grouped into separate categories we devised based on the elements commonly found in the SQL select statement – tables names, attributes, prefixes, symbols, aggregate functions, and a miscellaneous category for specific instances, such as when a proper noun needs to be referred to in the solution. While error groupings are based on the various elements of an SQL statement, the notion can be extended to most other formal languages.

SQL statements are generally made up of three clauses - *select* – *from* – *where*. Errors can occur in any of these clauses. The basic aspects of the language consist of input elements (*from*), a computation (usually based on a condition - *where*), and an output

element (*select*). While various errors can be made at each part of the formal language sentence the nature of the sentence makes it relatively easy to partition and analyse.

The tutoring system is currently being extended to include a multi-level error categorization scheme. Errors types will be further grouped according to the six fundamental SQL select statement clauses, i.e. select, from, where, group by, having, order by. Thus, errors will be classed as being syntactical or semantic, of a certain type (such as being concerned with table names), and occurring in a particular clause. Errors will also be assigned a severity level – some mistakes indicate more fundamental student difficulties than other errors made. By using a multi-level scheme, the system will be able to pinpoint mistakes and accompanying misconceptions (which may suggest pragmatic errors) in a more precise manner.

3.3 Interface

All interaction with the student is via the interface. There are a number of primary tasks (or questions), based on a particular SQL concept, for the learner to attempt (Figure 2). Central to each task is the testing of the student on a combination of error types; this is in a guided tutorial format.

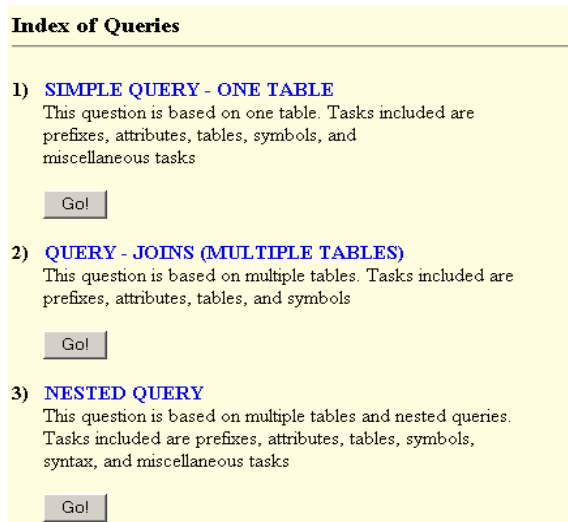


Figure 2. Question index

Having selected a task to attempt, the student is shown an initial input screen (as shown in Figure 3). The selected task is described in English. The student must submit his SQL answer to the given task, by typing into a dedicated text area, in order for his proposed solution to be executed. Initially the statement is used to query an actual database of relevant tables. The submitted solution is then evaluated by the correction model through a series of steps whereby patterns in the student’s solution are matched with patterns in the stored idea solution. A graphical representation of the student’s submission is displayed at this stage to aid clarity, and continues to be displayed until the user moves on to the next task.

Various items of data are produced by the correction model. The most fundamental piece of information notes if the proposed solution is deemed to be correct or incorrect. Other important data relates to errors that may have been made, such as type and location.

All such correction information, along with the student’s preferences, is stored and arranged by the student model. This model is updated for each student every time they submit an answer, regardless if submissions are made during one session or over a range of sessions.

The pedagogical model analyses the information stored by the student model, and so it plays an important role in the overall system. It is concerned with three main functions – *feedback*, *assessment*, and *guidance*.

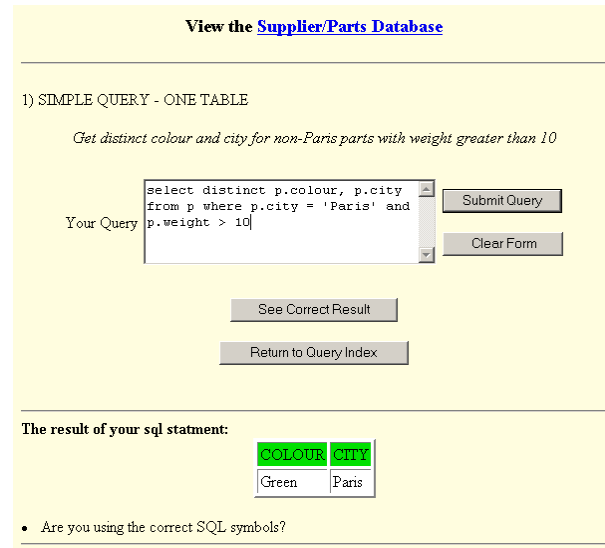


Figure 3. Input screen with hint displayed

3.4 Feedback

The Oxford English dictionary defines feedback as information on some activity or system return on input. In the educational context in question here, feedback is immediate advice given to a student based on solutions previously submitted by the student. Its objectives are self-reliant learning and the achievement of competency in a domain. It may be synchronous or asynchronous i.e. it may be returned immediately or returned after a period of time.

Our system provides increasing degrees of feedback support by offering a three-tiered synchronous feedback strategy, comprised of error flagging, hints, and partial solutions.

- *Error flagging*, which the system automatically displays, is a form of integrated feedback. It shows the user if their submission is deemed to be correct or incorrect. A syntax error results in a system error message being created by the actual Oracle database in use. This error message is amended for the learner or novice user, so that the student can more easily comprehend it. It is then printed on screen, integrating Oracle into the learning environment, in keeping with a realistic setting. Should the submission be semantically incorrect the student may choose to view hints, partial solutions, or both.
- *Hints* indicate the part of the query that is causing an error. This aims to guide the student towards the solution, without explicitly telling him the correct answer. He is encouraged to rethink and restructure his proposed solution in the

context of the given hint. In the example shown in Figure 1 the student has used the “equals” symbol instead of the “is-not-equal-to” symbol. The corresponding hint displayed is “Are you using the correct SQL symbols?”

- *Partial solutions* give the student precise advice. However, while this advice is directly related to the incorrect input sentence, it is not so specific as to tell the student the complete answer. He must judge how to use the partial solution in his answer. Regarding the example in Figure 1, the partial solution displayed would be “The question specifies non-Paris parts. Therefore, you need to use the <> symbol.”

Each type of feedback is displayed when the user selects a particular option. In keeping with guided discovery, it is not compulsory to view hints or partial solutions. Additionally, allowing the student to choose the level of feedback may result in fewer negative motivational consequences [5]. Should the student find he is unable to successfully complete the given task, with or without the aid of feedback, he may choose to view the correct answer.

3.5 Assessment and Guidance

Other SQL tutors provide a feedback feature of varying degrees [1][7]. As far as we are aware, however, there is no optional personalized guidance element available to the student.

The term guidance is defined by the Oxford English dictionary as help or advice. We extend this definition to describe guidance as offering the student advice and recommendations for areas to concentrate on, based on data collected in a previous determination. This may be a single determination (i.e. a single session), or data collected over a period of time (over a number of sessions). When teaching a classroom-based module, a lecturer’s advice is commonly based on the results of continuous assessment or coursework. Ideally an online tutoring system should also assume this role.

- You are having a small number of problems with selecting correct attributes
- You are having a lot of of problems with aggregate functions
- You are having a moderate amount of problems using SQL symbols

These results have been analysed, and a set of practise queries have been selected for you

1) SIMPLE QUERY - ONE TABLE

This question is based on one table. Tasks included are prefixes, attributes, tables, symbols, and miscellaneous tasks

Go!

2) QUERY WITH AGGREGATE FUNCTIONS

Supplier-Parts Database
This question is based on aggregate functions and a single table. Tasks included are attributes, tables, and aggregate functions

Go!

3) QUERY WITH AGGREGATE FUNCTIONS

Supplier-Parts Database
This question is based on aggregate functions and a single table. Tasks included are prefixes, attributes, tables, symbols, aggregate functions, and miscellaneous tasks

Figure 4. Portion of guidance feature

Information about the student is recorded as he works through the various tasks available on the tutoring system. This information

consists primarily of the number of attempts made for each task along with the amount and category of any errors made for those attempted tasks.

The system uses the data obtained by the assessment element in order to formulate personalized guidance for the student (Figure 4). This data is divided into various categories based on the types of error the student has made. Also, the number of errors and attempts the student has made for each error type weights it. Guidance includes explicitly telling the student the level of difficulty they are having with each error group, be it low, moderate, or high. The system makes this evaluation by studying the weighted information available for each student. They are then presented with a list of tasks to practise, from which they may choose to complete any number or combination. Guidance is weighted according to the amount of errors made. Making a large number of mistakes within a certain error group results in more tasks based on that group being suggested for practise.

It is not compulsory for the student to attempt the extra set of tasks. Each task is described in full to allow the student to decide which they wish to attempt.

A further improvement to the guidance element of the tutoring system can be achieved through the use of data mining or, more specifically, web usage mining. Chang et al. [3] define data mining as being the process of extracting information or knowledge from a data set for the purposes of decision making. We wish to achieve this using the information stored in the access logs created by the web server. In order to analyse this raw data in order to discover learning and training patterns, we will adopt a three-tier architecture [11], which consists of data gathering and pre-processing, pattern discovery, and pattern analysis. By dynamically assessing the previous actions of the student, for example the SQL topics he has recently browsed, personalization of feedback and guidance can be enhanced, and fading of scaffolding can be further supported.

4. EVALUATION

An evaluation was conducted to assess student acceptance and usage of the system. This was carried out over a period of time by surveying students, studying web logs created by the server, and analyzing student results.

4.1 Usage

It was noted that high usage occurred during the week, particularly in the late afternoons and evenings. However, a significant amount, 20%, of usage was observed on weekends. In a traditional classroom setting, a tutor would not typically be available on weekends. With an online tutoring system, however, the student may continue with coursework when they wish.

Students expressed the opinion that feedback is the most important feature of an online system; we have found that 84% of students surveyed have used the scaffolding provided. Feedback and guidance are integral parts of this scaffolding, and should be faded over time.

4.2 Student opinion

Students taking the Introduction to Databases module were surveyed to determine their attitude towards online learning as opposed to traditional classroom based learning [9]. In excess of 85% appreciate online teaching and skills training as part of an

on-campus degree programme. Students indicated that key advantages to the system include its constant availability and the self-paced learning opportunities.

When asked for their opinion regarding traditional labs versus online labs, the students surveyed did not greatly prefer one over the other. This indicates that both types of lab delivery are accepted.

4.3 Student results

Student performance is a substantial part of a degree programme. Over the course of the last four years there has been a 2% increase per annum in the marks obtained by students in SQL exams [9]. During this time the system has been improved; other factors have remained constant.

5. CONCLUSION

The popularity of online tutorials is increasing as university class numbers expand and the availability of the Internet grows. Online tutoring however can be a complex task. Often learning through technology results in the computer simply acting as a substitute for a textbook. In order to maximise the potential offered by technology, online tutoring systems should reflect the role of the teacher rather than the textbook. For example, it would be beneficial to offer feedback to the student when they are having difficulty. A further benefit would be to note the areas the student is excelling in or having problems with and offer advice based on this.

Formal languages are particularly suited to automated tutoring. Their structure makes the expressions in these languages relatively easy to analyse and are often easy to define. Therefore, the comprehensive automated tutoring of a formal language becomes a realistic goal.

The system we presented here is an automated tutor for a database skills training environment. It aims to help students develop their SQL skills and is focused on training rather than knowledge acquisition. SQL is a formal language and so it is suited to automated tutoring. As with most formal languages errors can be easily classified as syntactical, semantic, or pragmatic, and thus can be analysed.

Our strategy for the automated tutoring of a formal language, SQL, is based on the virtual apprenticeship model. As with a traditional apprenticeship the student is engaged in a learning-by-doing process. The master observes and offers assistance when necessary. The automated tutoring system presented uses attempts to emulate this using scaffolding in the form of feedback and guidance. In keeping with guided discovery the student may view increasing levels of feedback, namely error flags, hints, and partial solutions.

Typically the student will attempt a number of tasks over one or more sessions. Information about each attempt and any errors made are stored. This is retrieved and used if the student wishes to obtain dynamically created personalized guidance.

Analysis of student activities and surveying of students have shown this automated tutoring system to positively influence exam results, and students have accepted it equally with classroom based lectures. The tutoring system presented here

attempts to move beyond teaching the basic constructs of SQL alone. By encouraging the student to repeatedly practise the use of certain concepts he will gradually learn the pragmatics of the language, such as rules and conventions.

6. ACKNOWLEDGEMENTS

Our work was supported by the Dublin City University Teaching and Learning Fund.

7. REFERENCES

- [1] Bhagat, S., Bhagat, L., Kavalan, J., Sasikumar, M., Acharya: An Intelligent Tutoring Environment for Learning SQL. In *Proceedings of the Vidyakash 2002 International Conference on Online Learning*, Mumbai, India, 2002.
- [2] Boyle, T., *Design for Multimedia Learning*, Prentice Hall, Europe, 1997.
- [3] Chang, G., *Mining The World Wide Web*, Kluwer Academic Publishers, 2001.
- [4] Collins, A., Brown, J. S., & Newman, S. E. Cognitive apprenticeship: Teaching the crafts of reading, writing, and mathematics. In L. B. Resnick (Ed.), *Knowing, learning, and instruction: Essays in honor of Robert Glaser* (pp. 453-494). Hillsdale, NJ: Lawrence Erlbaum Associates.
- [5] Corbett, A., Anderson, J., *Locus of Feedback Control in Computer-Based Tutoring: Impact on Learning Rate, Achievement and Attitudes*. In *Proceedings of the SIGCHI conference on Human factors in computing systems (CHI'01)*, New York, USA, 2001.
- [6] McLoughlin, C., Winnips, J. C., and Oliver R., Supporting Constructivist Learning through Learner Support On-line. In *Proceedings of the World Conference on Educational Multimedia, Hypermedia and Telecommunications EDMEDIA 2000*, Montreal, Canada. 2000, pp 674-680.
- [7] Mitrovic, A., A Knowledge-Based Teaching System for SQL. In *Proceedings of the World Conference on Educational Multimedia, Hypermedia and Telecommunications ED-MEDIA/ED-TELECOM'98*, AACE, Freiburg, 1998, pp. 1027-1032.
- [8] Murray, S., Ryan, J., and Pahl, C., A tool-mediated cognitive apprenticeship approach for a computer engineering course. In *Proceedings of the IEEE International Conference on Advanced Learning Technologies ICALT'03*, 2003.
- [9] Pahl, C., Barrett, R., and Kenny, C., Supporting Active Database Learning and Training through Interactive Multimedia. In *Proceedings of the Annual conference on Innovation and Technology in Computer Science Education ITiCSE'04*, Leeds, UK, ACM, 2004.
- [10] Wenger, E., *Artificial intelligence and tutoring systems : computational and cognitive*, Morgan Kaufmann Publishers, 1987
- [11] Zaïane, O., Luo, J., Towards Evaluating Learner's Behaviour in a Web-Based Distance Learning Environment. In *Proceedings of the IEEE International Conference on Advanced Learning Technologies ICALT01*, pp 357-360, Madison, WI, USA, 2001.