Evaluation of an Intelligent Tutoring System used for Teaching RAD in a Database Environment

Silviu Risco, Jim Reye

Computer Science Discipline, Faculty of Science and Technology
Queensland University of Technology (QUT),
GPO Box 2434, Brisbane, QLD 4001,
Email: {s.risco, j.reye}@qut.edu.au

Abstract

This paper presents an evaluation of the Personal Access Tutor (PAT), an Intelligent Tutoring System (ITS) for Learning Rapid Application Development (RAD) in a database environment. We first give an overview of Microsoft Access, the environment that PAT uses. After describing related work in the field, we discuss the architecture of PAT and the services that PAT offers to the students, together with a short introduction of how students use PAT. After presenting the evaluation methodology, the results of a summative evaluation are discussed. Additional evaluation using data gathered from students by PAT is analysed as a pre-post test. The paper concludes with a summary and describes further work.

Keywords: ITS evaluation, Intelligent Tutoring Systems, Student Modelling.

1 Introduction

1.1 Microsoft Access overview

Microsoft Access (aka Microsoft Office Access) is a Relational Database Management System (RDBMS) developed by Microsoft Corporation. From this point on, to simplify the text, Microsoft Access will be called Access. Access is the most widely used Windows desktop RDBMS.

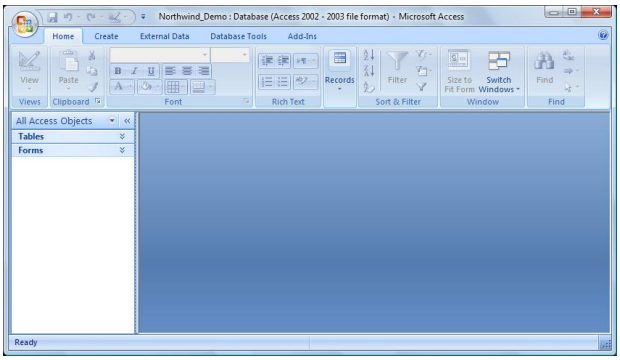

Figure 1: Microsoft Access - graphical interface

Several versions of Access have been developed by Microsoft Corporation. The latest version is Access 2010. PAT was initially developed for Access 2003 and later ported to Access 2007 and 2010. PAT can now be installed on any of the combinations of Access 2003 - Access 2010 and Windows XP, Vista or Windows 7.

Access is based on the Microsoft Jet Database Engine and provides a graphical user interface to create and use databases - see Figure 1. Through the graphical interface, the users can create several types of objects such as forms and reports to easily interrogate or update the database. Figure 2 and 3 show examples of forms and reports created in Access.

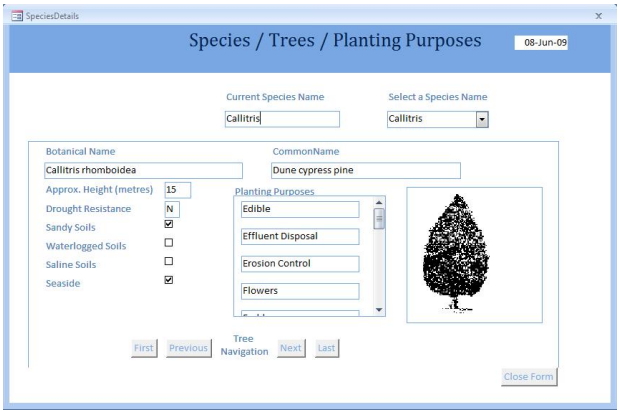

Figure 2: Microsoft Access - example of a form

Forms are objects used to enter, view or edit records in a database; reports are formatted printouts of the content of one or more tables or queries from a database (Adamski & Finnegan 2008).

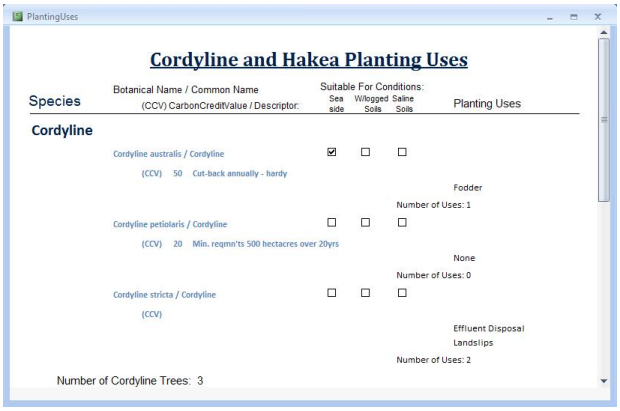

Figure 3: Microsoft Access - example of a report

Access users can use built-in wizards to create simple

forms and reports, but these have restricted layouts and functionality. While using the wizards provides an easy start for users who have just started to learn Access, the wizards cannot be used to automatically create more complex and more advanced forms and reports. Such forms and reports must be manually created, although the wizards can be used to create a basic version, as a starting point. Users can also design forms and reports from scratch.

1.2 Teaching RAD using Access

"Databases" is a first year subject at Queensland University of Technology (QUT). While the first part of the subject covers SQL, the second part provides the opportunity to use a commercial RDBMS for Rapid Application Development (RAD) while applying knowledge learned about SQL. Because it is a widely used RDBMS, students learn how to use Access. Students have weekly practicals where they are required to solve exercises related to that week's topic.

For the Access part, students learn how to create queries, forms and reports. First, they have to create simple forms and reports using the wizards, then they are required to improve the initial forms and reports, adding more functionality and improving their appearance.

1.3 Using PAT as an additional tool

PAT, an ITS for Access, aids students' learning, complementing the lectures and practicals. PAT is freely available for QUT students. Students can use PAT both during the practicals or at home, in their own time and place.

It can be used to learn how to create forms and reports in Access, using PAT's built-in exercises. It can also be used for assistance when working on assignments for the Databases subject.

1.4 Related work

Although of a limited number, other ITSs for learning about databases exist. These ITSs focus on teaching database domains such as Structured Query Language (SQL) and Database Design. SQL is a database language program designed for data management and manipulation for relational database management systems. Database Design is the process of creating a model of the information that will be held in the database. In this section we briefly describe some of these ITSs.

DB-suite (Mitrovic et al. 2004, Mitrovic et al. 2008) consist of three web-based intelligent tutoring systems in the area of databases:

- SQL-Tutor: teaches the SQL query language;
- NORMIT: a data normalization tutor; and
- KERMIT: teaches conceptual database modelling using the ER model.

These tutors are constraint-based tutors (Mitrovic & Ohlsson 1999,Mitrovic & Weerasinghe 2009, Ohllsson 1992). In the case of constraint-based tutors, the system analyses the student's solution, checking if any constraints from the domain model are violated. The constraints are both for correctness and completeness. If a solution does not violate any constraints, then the solution is considered correct and complete.

The DB-suite tutors are designed to be used as an additional tool, to complement classroom teaching.

SQL Tutor (Mitrovic 1998, Mitrovic & team 2008) is a constraint-based ITS for students learning SQL. When using the tutor, the students have to complete SQL statements satisfying the given requirements.

The system contains definitions of several databases and a set of problems, together with their ideal solutions. The domain model of SQL Tutor contains more than 700 constraints.

Kermit (Suraweera & Mitrovic 2002, 2004) is an ITS for teaching Database Design using the Entity-Relationship (ER) data model. The students have to create an ER diagram based on the requirements given by the system. Kermit provides feedback to the students by request only.

The students can ask for a hint or can ask for the solution to be evaluated.

The feedback level is automatically increased each time the student asks for help, up to the *hint* level. Kermit contains over 200 constrains, both syntactic and semantic constraints.

Normit (Mitrovic 2002) is an ITS for students learning Database Normalisation. Normalisation is part of the database design. Data normalization is concerned with data optimisation, to minimise redundancy. Because Database Normalisation is a procedural task, the students have to follow a strict sequence of steps to solve the problem and the system does not need to store a correct solution.

The domain model of NORMIT contains more than 80 constraints (both syntactic and semantic) to check the student's solution.

The hints have only two levels: a general hint and a more detailed hint. On the first time of violating a constraint, the system presents the general hint. When the rules are violated again, the more detailed hint is presented.

Acharya (Bhagat et al. 2002) is an web-based ITS for learning SQL. Acharya only analyses SQL for database querying, not updating.

This ITS uses Java servlet technology on a web-based front-end and PostgreSQL as a back-end.

Acharya contains a student module and a pedagogical module. The architecture has two separate databases, one for the student model, and the other one for the rest of the models - including the problems and their solutions. The student model contains general information about the student, history of concepts learned, with a confidence factor (the system's belief that the student acquired the concept), knowledge level and number of hints received.

In contrast to SQL Tutor, Acharya uses a real RDBMS to run the students' solutions and the result of the query is returned back to the student - if the query is correct. However, the students must still use the tutor's interface to write the parts of the select statement.

Acharya stores in its student model general information about the student and history of information about the concepts learned (Bhagat et al. 2002). The concepts learned are recorded with a *certainty factor* which is a measure of Acharya's belief that the student has acquired the concept. In addition, Acharya also records a knowledge level and the number of hints asked by the student.

Acharya can propose problems to the student based on pre-requisite relations. If the student's solution is correct, the result of the SQL is displayed. If the student's solution has errors, the most basic ones are addressed.

Showing the results of the query the students created as returned by a real RDBMS is very beneficial for the students as they can see exactly the result of their work.

SQL Lightweight Tutoring Module (SQL-LTM) (Dollinger 2010) is a system that can provide semantic feedback on SQL statements, pointing out their logic flows, even if they are syntactically correct. It can detect most conceptual errors that SQL learners can make (Dollinger 2010).

SQL-LTM is integrated with a Web based AJAX universal query tool called AJAX Enabled Query (AEQ) (Dollinger et al. 2009).

SQL-LTM consists of two modules: a query parser which converts the SQL query into an XML representation, and an analyser which compares the test query provided by the student against the reference query created by the instructor - also provided in an XML representation.

One of the difficult issues in analysing SQL queries is the possibility that the students can provide solutions that even though are different than the optimal solution, they can still be syntactically and semantically correct. The analyser recognises the semantic equivalence of such queries and provides recommendations on how to get to the expected solution.

Similar to Acharya, SQL-LTM uses a real RDBMS to run the queries from the students' solution. However, SQL-LTM does not have a student model nor does it keep track of student's history; hence the system cannot individualise the feedback. For the same error, different students receive the same advice.

2 Personal Access Tutor (PAT)

2.1 Architecture and components

An ITS architecture usually contains a simulation module which is used to replicate the real environment that the student is learning about. The architecture of PAT is different from this because it uses the real working environment (Access) instead of a simulation module.

Figure 4 [1] presents the architecture and main components of PAT.

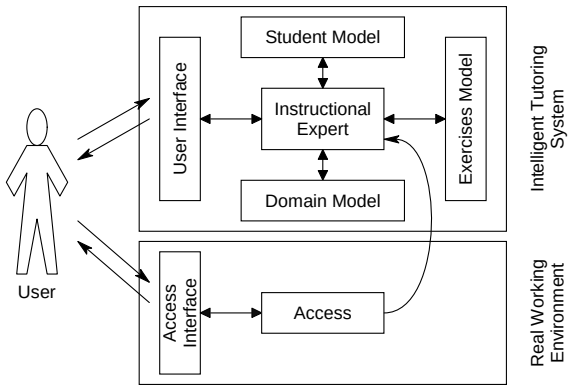

Figure 4: PAT's architecture

The *Domain Model* represents the knowledge about the domain to be taught, knowledge that an expert in the field should know. It is the foundation for the entire knowledge base. Because PAT's focus is on helping students to learn how to create forms and reports, all the objects (together with their properties) that can be created in a form or a report are present in the Domain Model.

[1] based on the general architecture of an ITS presented by Burns & Parlett (1991)

The *Student Model* contains the system's beliefs regarding the student's knowledge of the domain (Holt et al. 1994) and additional information about the student, such as personal characteristics and learning style (Beck et al. 1996). In PAT, the Student Model includes information about student's preferences for learning from diagrams or text, their interests in the subject and in several topics from the domain. Every time PAT analyses the student's solution, the Student Model is updated with new information.

The *Instructional Expert*, based on knowledge both about the domain and the student, diagnoses the student's attempted solution and provides individualised feedback. As part of the Instructional Expert, the Tutoring Model contains information about teaching the domain such as tutoring goals and hints for students. The Tutoring Model must be able to take advantage of the information provided from the Student Model (e.g. student's learning style and personal characteristics). The Instructional Expert in PAT is based on principles from the Minimalist Framework for designing instructional materials for computer users (Carroll 1990); the GOMS model (Card et al. 1983); and Andragogy, "the art and science of helping adults learn" (Knowles 1980, p. 43) - a student centred approach for adults.

Access / Access Interface: because PAT is implemented as an add-in for Access, the student can utilise PAT from within Access. After installation, PAT appears as a new group in the Access's ribbon. In this way, the student can actually work on each exercise, test their solution and receive feedback from PAT without leaving Access's main window.

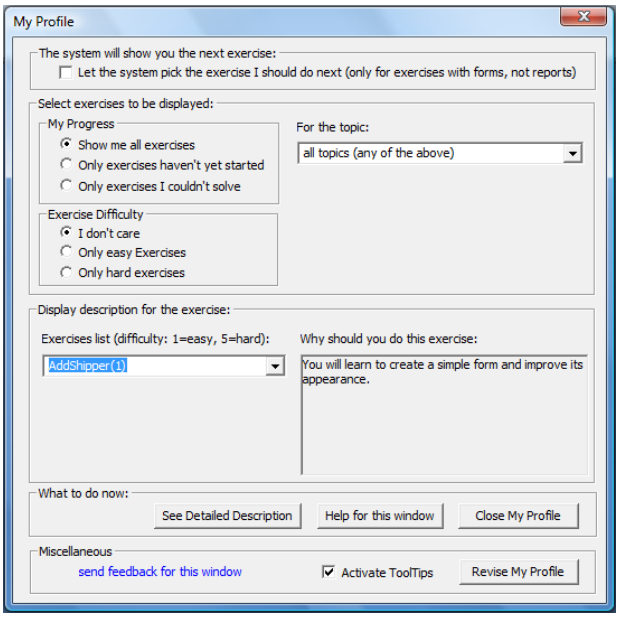

Figure 5: PAT's My Profile window

2.2 Services that PAT offers

VanLehn (2006) presents a global approach to ITSs behaviour. Based on the concepts of *Task* (a multi-minute activity that can be skipped or interchanged with other tasks) and *Step* (multiple user interface events that together can complete a task), an ITS is presented as having 2 loops: the outer and the inner loop. The *outer loop* is responsible for the task selection, similar to the "elaborative function" identified by Self (1987). The *inner loop* consists of the steps inside the task: assessment of knowledge (diagnostic

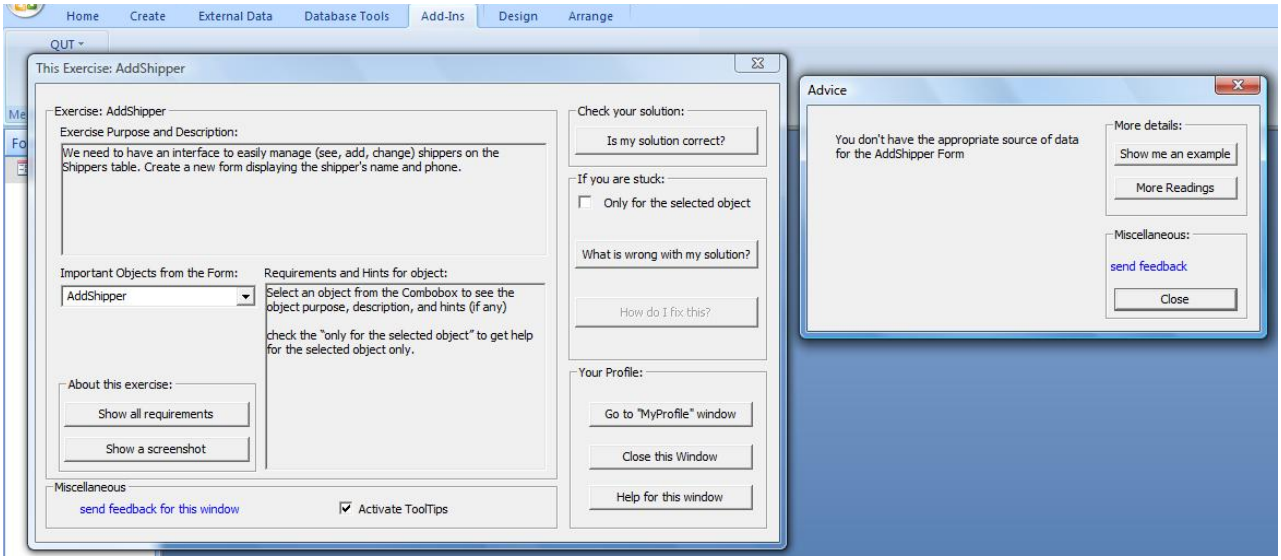

Figure 6: PAT's This Exercise window

function), feedback and hints (corrective function), etc.

PAT's *outer loop* (task selection) gives the student two choices: the student can select an exercise or can ask PAT to suggest the next exercise (or the first exercise, if the student did not try any exercises yet). The selection can be narrowed down by specifying the difficulty of the exercises or whether the exercise was previously attempted. The exercises are categorised by topics (and subtopics) to be learned and contain (beside the exercise description) the exercise difficulty and what the user will achieve by completing the exercise. Figure 5 shows the *MyProfile* window, which is the interface from where the user can choose which exercise to do next.

PAT can propose an exercise based on the information from the student model and considering the following principles:

- The exercises should contain topics not mastered yet by the student.
- The topics not known yet should help to complete broader teaching concepts.

PAT's *inner loop* relates to the steps within a task and can be grouped in two main categories of services offered to the user: step generation and step analysis. While the step generator is about what the user should do next, the step analyser is responsible for other actions such as answering the question "Is it correct?", and other types of feedback. In PAT, the user can access these services from the *ThisExercise* window (left hand side of Figure 6).

- Is my solution correct?
- What is wrong with my solution?
- How do I fix this?

Is my solution correct? gives the user an overall presentation of what is correct and what is incorrect with their solution. PAT displays the "Traffic Lights" image (Figure 7) where there is a row for each important task in the exercise. A green light means the task is correctly done, while a red light means the task is completely wrong or missing. Because a task can consist of a series of steps, a yellow light means that *only some* steps are correct, not *all* of them.

What is wrong with my solution? gives minimal feedback - it only describes what the error is. *How do I fix this?* gives the user feedback about the actions that should be done to correct an error. This type of feedback is not enabled the first time the user asks for help. Generally, the user has to first choose *What is wrong with my solution?*, and only after that can they ask for help on how to fix the error. In the case where PAT detects that the user will not benefit from receiving a general (vague) hint, a more specific hint will be provided. That could be about how to fix the error, rather than what is wrong.

The feedback for the last two services above is grouped on several levels of specificity, starting from a general hint and leading to more specific hints. However, not even the most specific feedback gives away the correct solution because PAT can be used even for assignments.

In addition to the services described above, PAT can display a diagram depicting the context of the error (Figure 8). PAT can also give users references to readings related to the topic where the error occurred. The additional readings are from lecture notes, recommended books or lecture slides.

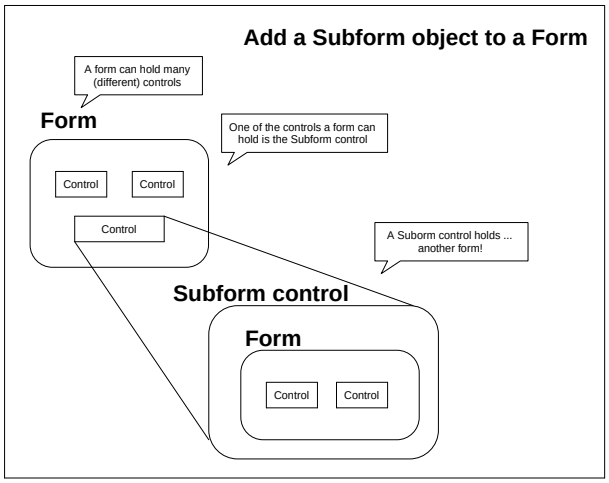

Figure 8: Example of a diagram that PAT displays

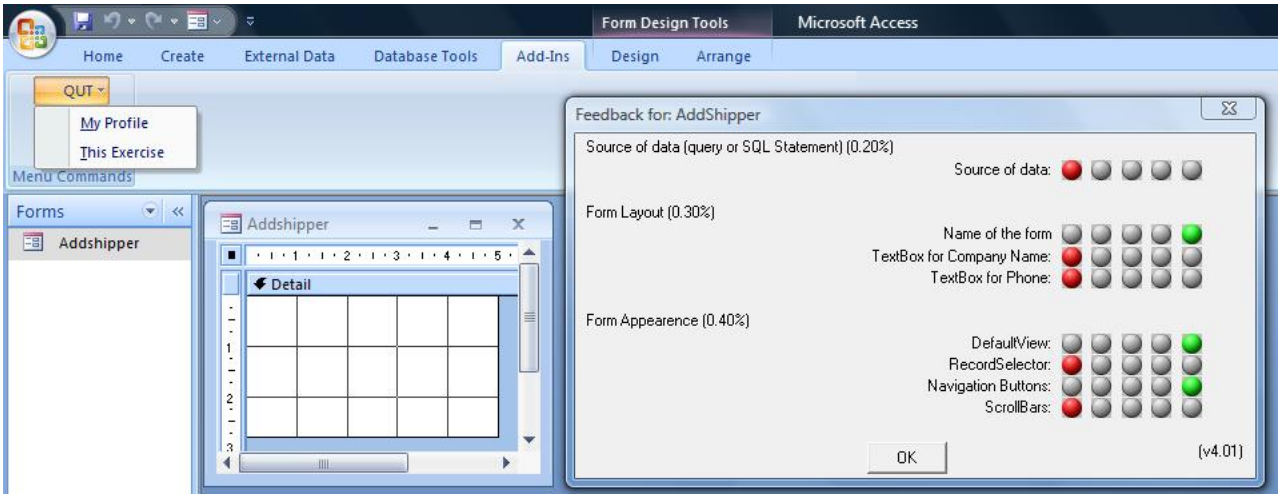

Figure 7: PAT's This Exercise window

3 How PAT is used

PAT is freely available to QUT students enrolled in the Database subject and can be downloaded from the learning content management used at QUT. PAT can be installed on home computers or laptops, as well in computer labs. However, to take maximum advantage of the Student Model, installing PAT on students' laptops or home computers is recommended.

There are two ways in which PAT can be used:

- go through enough exercises to cover (and master) the entire curriculum; and
- use the system mainly to work through the assignments.

The first option is more suitable for students who do not have any previous knowledge working with Access or other RDBMS. The second option is preferred by students with some previous knowledge with Access or students with less interest in the subject. The two options of using PAT are presented next.

3.1 Best way to work with PAT

The best way to work with PAT is to take full advantage of the Student Model that PAT will update during the student-system interaction. This implies both following the exercises that PAT will propose and also keeping the Student Model. While the first only needs the student to ask PAT for the next exercise instead of manually select one, the second requires the student to consistently use the same installation of PAT - same computer. In this way, every exercise that the student will attempt will be recorded in the Student Model.

As the student works on the exercise that PAT proposes, (s)he can check if their solution is correct. By displaying the traffic lights, the student has a good indication of what (if anything) is wrong with their solution. If the error is not obvious, the student can ask PAT what is wrong.

Before asking for more help, the student should first check the helping diagrams and more readings section, trying to discover by themself what to do next. Only if the student is still stuck after more readings, should they ask PAT for hints on how to fix the problem.

3.2 Using PAT for assignments

PAT is released to the students at the same time as the assignments. Because there are three Access assignments, there are three releases of PAT, which include (in addition to the existing exercises) the requirements and solutions for the assignments. The solutions are kept hidden from student view, via encryption and other security mechanisms.

Students' previous experience with Access or other RDBMSs can vary significantly. We understand that some students have already used Access. Therefore assuming that they will go through the entire process described in the previous section would be wrong. In these circumstances, we allow the students a different approach. They can try to use only the *Is my solution correct* feature, and keep working without spending time on more readings - assuming that they have the required knowledge.

4 Initial evaluation of PAT

Evaluations were conducted with students who used PAT. A survey was used to provide both qualitative and quantitative data.

Students enrolled in this subject have diverse backgrounds and their previous experience with Access (if any) varies significantly. For an accurate interpretation of students' answers, we distinguish between students who have used Access before and those who haven't. Another important aspect that has to be considered is how much they used PAT during the semester. If some students only seldomly used PAT they will not be in the same category as students who used PAT extensively.

4.1 Objectives and Methodology

Iqbal et al. (1999) suggested that the evaluation method for an ITS should be chosen by what is being evaluated (the entire system or only a part of the system) and the number of available students. We wanted to analyse PAT as a system and we had 185 students enrolled in the Databases subject in second semester 2008. We used questionnaires as an exploratory research method. During one of the lectures, the students present in the lecture theatre were asked to fill in a two pages questionnaire. Of 185 students enrolled in the subject, 84 responded to the

questionnaire. However, only 51 of the students answered all questions.

The objectives for the evaluation were:

1. Get students' backgrounds.
2. Does PAT's architecture (receiving help when solving real problems) make learning easier?
3. Is the feedback provided helpful for students? Is PAT offering enough types of help?
4. Is PAT accepted by students?

4.2 Conclusions for Summative Evaluation

This section describes the summative evaluation of PAT. We present the results from the evaluation with students, then the results from the evaluation with teaching staff. The results from both students and teaching staff are summarised below, grouped by the evaluation's objective. Because objective 4 measures the overall rating of PAT, it is presented first.

Objective 4 - Is PAT accepted by students and teaching staff?

As shown in Figure 9, we can see that both students and staff found PAT helpful. Furthermore, from questions 9 and 10 for students and question 6 for staff we see:

- the majority of students and staff members think that PAT is easy to use; and
- students would like to have software similar to PAT in other subjects.

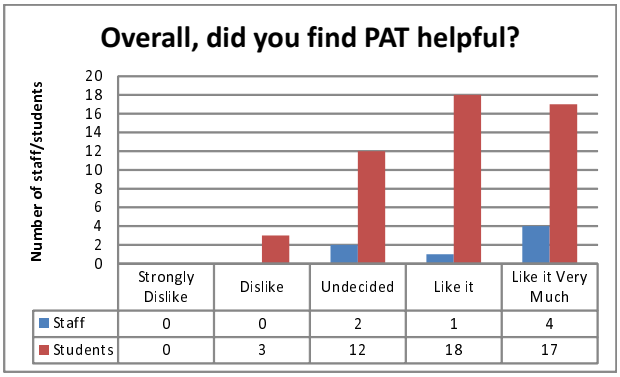

Figure 9: Students and teaching staff answers for questions 11 and 7.

Objective 1 - Students and staff members' background

Two thirds of the students enrolled in the subject have used Access before, with more than 50% of them being *Somewhat Confident* in using it. Interestingly, the answers to the questions between the two groups (students that used Access before and students that didn't use Access before) are similar in the majority of the questions.

The question where the results are different are:

- Q7 - Which type of feedback did you like least?
- Q11 - Overall, did you find PAT helpful?

Students who used Access before were not really interested in additional materials such as *Diagrams*, *More Readings* and not even in *How to Fix* while students who haven't use Access before did not like *More Readings* and *What is Wrong* but were happy with *Diagrams* and *How to Fix*.

Although one might expect that PAT would be more useful for beginners (students who had not used Access before), the results show that 56% of them answered that they like or like very much PAT and 38% answered that they are undecided. In contrast, 76% of the students that used Access before answered that they like or like very much PAT with only 18% undecided.

Objective 2 - Does the approach of having PAT embedded in Access make learning easier?

Both students and teaching staff considered that using PAT directly from Access, while using the real software (no simulation) to work on real problems, is very useful. Students' answers for question 10 can also be seen as a confirmation - the students would like to have software similar to PAT in other subjects.

Objective 3 - Is the feedback provided helpful for students? Does PAT offer enough types of help?

Questions 6 and 7 show students' preferences for the types of feedback provided. The *Traffic Lights* (error indication) are by far the most liked type of feedback see Figure 10. While the students who used Access before were looking more for a simple way of indicating what is wrong, the students who didn't use Access before were looking not only for an indication of "what is wrong" but also for an indication on the overall performance and progress.

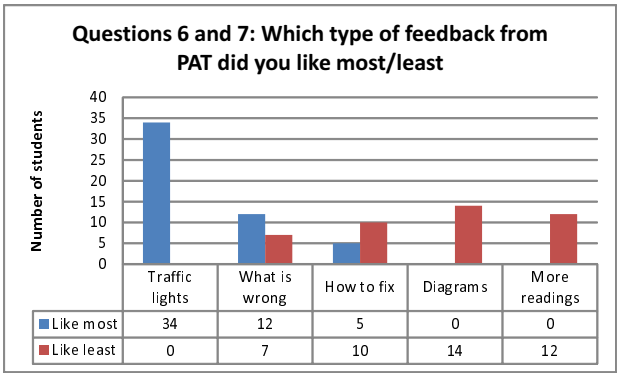

Figure 10: Type of feedback that students like most/least.

The least preferred types of feedback are *More Readings* and *Diagrams*: *More Readings* for the students who hadn't used Access before, while *Diagrams* for the students who had used Access before. Some teaching staff and students did not indicate any type of feedback as disliked (Question 7 for students and Question 4 for staff): "N/A - no particular dislike" or simply "none". The only suggestion for improvements from staff members (Question 3) was to not only have general diagrams describing the overall concept but also screen-shots from Access on how to solve some of the possible issues.

Some students were unhappy with the content of the feedback received. One possible explanation is the fact that PAT only gives hints, not the solution (correct answer) for assignments. However, future work could look at ways to improve the feedback.

5 Pre-Post test evaluation

In addition to the evaluation described in the previous section, we gathered measurements of the students' knowledge before and after using PAT. During the interaction between the student and PAT, PAT records every click on any of its interfaces, together with additional information about students solution

at that time such as: the current error, the advice given, etc. The information is recorded in another Access database containing the student model. When submitting the assignment, the students were asked to also submit this database containing their student model.

Because PAT helps with the assignments, we could not ask any of the students not to use PAT just to have a control group. For this reason, we evaluated PAT using the approach described by Woolf (2008, p. 191) as "C1. Tutor alone". This evaluation was conducted during the second semester of 2010, with a different set of students.

5.1 Information sources

To analyse the improvement engendered in students' learning while using PAT we have two available sources of information: data gathered by PAT during student-system interactions and the students' solutions to the assignments.

During each interaction between the student and the system, PAT will record not just the current session, the error that was addressed, the advice type, advice code, etc. (as part of the student model) but also the name of the dialog box and the button - any time the student clicks on one of those. The data is recorded in the database containing the student model. The information in this database allows us to see which topics were not initially understood (not known) by students while they were using PAT - either practicing on the helping exercises or working on the assignments.

Additionally, the students' solutions to the assignment provides us with information not just about the topics not understood by the students (not known), but also the topics shown as understood (known) by the students.

The first source of information (student-system interaction) was used to provide pre-test data, while the information from the second source (the assignments) was used to provide post-test data.

5.2 The student population

Of the 235 students enrolled in the Databases subject in the second semester 2010, 199 students submitted all the assignments and student models. Because using PAT was optional, the information received from some of these 199 students was insufficient for an accurate evaluation of the data for those students.

Possible criteria for selecting the relevant students (students that used PAT enough to provide useful data) are:

- number of interactions (clicks on PAT's interfaces) - a maximum of 1598 and a mean of 182.5;
- number of advice messages received - a maximum of 394 and a mean of 30; and
- number of sessions started - a maximum 151 and a mean of 11.

The first criterion above is a good measure of how much a student used PAT, in order to distinguish between significant and insignificant data. This criterion was used to determine the best set of data - that for the first 100 students, in descending order of their numbers of interactions.

5.3 The topics considered

From the 38 topics (object-properties) existing in the exercises used during both the pre and post tests, we selected the 10 most relevant ones based on the following criteria:

- the topic should require the student to set a correct value (i.e. not topics that can be easily generated by the wizards or using default values); and
- the topic should be important from a teaching perspective (i.e. some object-properties are more important than others).

5.4 Results of the pre-post evaluation

In Section 5.1, we explained that the data being used for pre-test purposes provides information about topics not known initially. Where students did not need help, we take it for granted that they already knew the topic. The post test data provides information about topics known and unknown.

From this data, we obtained the average number of topics known on the pre-test versus average number of topics known on the post-test. These values are shown in Figure 11.

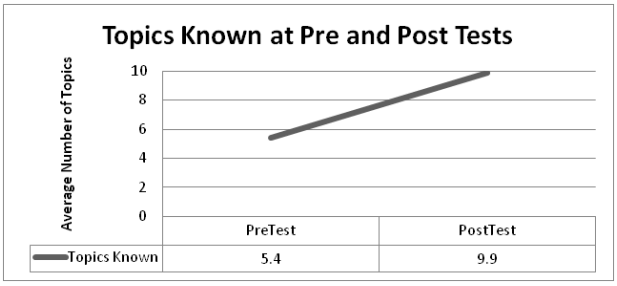

Figure 11: Topics known at pre and post tests.

On average, the number of topics known increased from 5.4 to 9.9, after using PAT, i.e. the average number of topics learned is 4.5. The mode is 4 topics learned, with a standard deviation of 1.87.

Woolf (2008, p. 191) lists for a "tutor alone" evaluation the questions that should be addressed:

- Do learners with high or low prior knowledge benefit more?
- Do help messages lead to a better performance?

To address the first question, we analysed the pre-post results for students with prior knowledge of Access versus students with no prior knowledge of Access. When a student starts using PAT for the first time, it asks the student if they are confident with using Access. The answer is recorded in the student model.

Based on this information, 55 students (out of 100 students analysed) had prior knowledge of Access - i.e. they answered "yes" to the question if they are confident with using Access. The results of the pre-post test for the two categories of students are shown in Figure 12

It can be seen from the graph that the students with no prior knowledge had a slight increase in the number of topics learned in comparison with students with prior knowledge of access. Students with no prior knowledge had an average of almost 6 topics learned compared with 5 topics learned by students with prior knowledge of access.

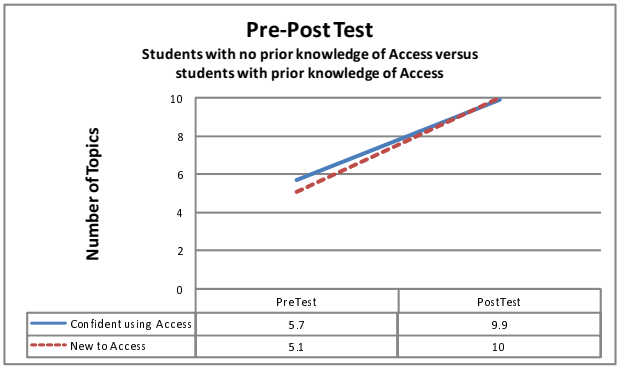

Figure 12: Students confident using Access vs students new to Access.

Regarding the second question that Woolf (2008) recommends that should be addressed, as we stated at the beginning of this section, we could not ask some of the students not to use PAT or to use a version of PAT without feedback messages or with different feedback messages. However, in Section 5.1, we have shown the students' opinion about messages received.

The distribution of number of students by the number of topics learned, is depicted in Figure 13.

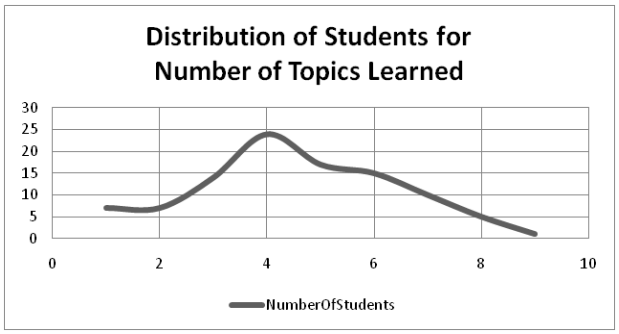

Figure 13: Distribution of students by topics learned.

The graph shows how many students learned 1 topic, 2 topics, and so on up to 9 topics - the maximum number of topics learned out of the 10 topics analysed. It can be clearly seen that the majority of them (about 70%) learned between 3 and 6 topics.

6 Conclusions and further work

The results of the evaluation showed PAT's usefulness for students' learning, as well as PAT's acceptance by both students and staff members.

These results also showed differences between students who had used Access before starting the subject, and those who had not. The differences were in the way the students used PAT and in the type of feedback they prefer. In addition, from the results of the evaluation it can be seen that the students would like to have ITSs similar to PAT for other subjects.

In addition, to show the improvement that PAT engenders in student learning, we used the data that PAT gathers as a pre-post test. The results from the test show that the students who used PAT had an average number of topics learned of 4.5 (out of the 10 most important topics analyzed), with a mode of 4 topics learned and a standard deviation of 1.87.

Because of PAT's modular structure, further enhancements can be made. The enhancements to PAT from which the students could benefit are:

- analyze (and provide help on) not just the correctness of the solution but also the readability and usability of the form or report;
- an open student model; and
- reports and statistics for teaching staff about students' learning performances.

Each of these is elaborated below.
A). When analysing the student's form or report, PAT analyses its correctness from a functional point of view. I.e. is the form or report producing the correct data? From a human user perspective though, the readability and usability of the form or report could also be analyzed.

From a readability point of view, the objects in the form should have the same size, should be aligned, and should be grouped by their meaning or function. From an usability perspective, the fields should be displayed in the most meaningful order i.e. in the same order in which the data will be entered - first name, last name and address; not last name, address, and only then the first name.

This approach would involve PAT analysing other aspects such as the relative position and size of each of the objects in the form or report.
B). An open student model would allow the students to check their profiles. This would help the students' learning by facilitating metacognitive processes and providing them with an opportunity to reflect on their progress.
C). A benefit to the teaching staff would be to generate reports and statistics about students' learning performances. Aggregated data collected by PAT could help the teaching staff identify topics that are hard to learn, suggesting areas for future improvements.

References

Adamski, J. J. & Finnegan, K. T. (2008), *New Perspective Microsoft Office Access 2007 - Comprehensive*, Thomson Course Technology.

Beck, J., Stern, M. & Haugsjaa, E. (1996), 'Applications of ai in education', Web Page; http://www1.acm.org/crossroads/xrds3-1/aied.html. last visited 05 Oct 2011.

Bhagat, S., Bhagat, L., Kavalan, J. & Sasikumar, M. (2002), Acharya: An intelligent tutoring environment for learning sql, *in* 'Proceedings of Vidyakash 2002 International Conference on Online learning'.

Burns, H. L. & Parlett, J. W. (1991), The evolution of intelligent tutoring systems: Dimensions of design, *in* H. L. Burns, J. W. Parlett & C. L. Redfield, eds, 'Intelligent tutoring systems : evolutions in design', Lawrence Erlbaum Associates, Hillsdale, NJ, pp. 1–12.

Card, S. K., Moran, T. P. & Newell, A. (1983), *The psychology of human-computer interaction*, L. Erlbaum Associates, Hillsdale, N.J.

Carroll, J. M. (1990), *The Nurnberg funnel : designing minimalist instruction for practical computer skill*, MIT Press,, Cambridge, Mass.

Dollinger, R. (2010), Sql lightweight tutoring module - semantic analysis of sql queries based on xml representation and linq, *in* 'Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications 2010', AACE, Toronto, Canada, pp. 3323–3328.

Dollinger, R., Ford, R., Helf, B. & Reimer, K. (2009), 'Ajax enabled query tool the capstone experience', *Information Systems Education Journal* **7**(49). **URL:** *http://isedj.org/7/49/*

Holt, P., Dubs, S., Jones, M. & Greer, J. (1994), The state of student modeling, *in* J. E. Greer & G. McCalla, eds, 'Student modelling : the key to individualized knowledge-based instruction', Vol. 125 of *NATO ASI Series F: Computer and Systems Sciemces*, Springer-Verlag, pp. 3–35.

Iqbal, A., Oppermann, R., Patel, A. & Kinshuk (1999), A classification of evaluation methods for intelligent tutoring systems, *in* 'Software-Ergonomie 99, Design von Informationswelten, Gemeinsame Fachtagung des German Chapter of the ACM, der Gesellschaft fur Informatik (GI) und der SAP AG', Teubner, pp. 169–181.

Knowles, M. S. (1980), *The modern practice of adult education : from pedagogy to andragogy*, Cambridge Adult Education, New York.

Mitrovic, A. (1998), Learning sql with a computerized tutor, *in* 'Proceedings of the twenty-ninth SIGCSE technical symposium on Computer Science Education', ACM Press, Atlanta, Georgia, United States, pp. 307–311.

Mitrovic, A. (2002), Normit, a web-enabled tutor for database normalization, *in* 'Proceedings of the International Conference on Computers in Education ICCE', Vol. 2, Auckland, New Zealand, pp. 1276–1280.

Mitrovic, A., Suraweera, P. & Martin, B. (2004), 'Db-suite: Experiences with three intelligent, web-based database tutors', *Journal of Interactive Learning Research* **15**, 409–432.

Mitrovic, A. & team, T. I. (2008), Constraintbased tutors, *in* B. P. Woolf, E. Aimeur, R. Nkambou & S. Lajoie, eds, '9th International Conference on Intelligent Tutoring Systems', number (LNCS) 5091 *in* 'Lecture Notes in Computer Science', Springer-Verlag, Montreal, Canada, pp. 29–32.

Self, J. (1987), Student models: what use are they?, *in* P. Ercoli & R. Lewis, eds, 'Artificial Intelligence Tools in Education', Frasacati, pp. 73–86.

Suraweera, P. & Mitrovic, A. (2002), Kermit: A constraint-based tutor for database modeling, *in* S. A. Cerri, G. Gouarderes & F. Paraguacu, eds, 'Intelligent Tutoring Systems', Vol. 2363 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 377–387.

Suraweera, P. & Mitrovic, A. (2004), 'An intelligent tutoring system for entity relationship modelling', *International Journal of Artificial Intelligence in Education* **14**, 375–417.

VanLehn, K. (2006), 'The behavior of tutoring systems', *International Journal of Artificial Intelligence in Education* **16**, 227–265.

Woolf, B. P. (2008), *Building Intelligent Interactive Tutors: Student-centered Strategies for Revolutionizing E-learning*, Morgan Kaufmann.