http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 5 (5): 916-922, 2006
ISSN 1812-5638

Intelligent Tutoring System for Teaching Database to Sophomore Students in Gaza and its Effect on their Performance

Samy S. Abu Naser
Faculty of Computer Engineering and Information Technology,
Al-Azhar University Gaza, Palestine

Abstract: This study presents an Intelligent Tutoring System called DB-ITS for teaching database to sophomore students in computer science in Al-Azhar University. DB-ITS is designed as a guided learning environment and supports problem solving. An evaluation was carried out using DB-ITS to see if students performance in the database exams using DB-ITS is enhanced. The results indicated that the advantage of using DB-ITS over the conventional methods, the use of ITS was effective in improving students performance in exams and gender had no effect on the use of the ITS program designs in our study; furthermore, the removal of multimedia effects reduced the advantages of using ITS.

Key words: DB-ITS, Intelligent tutoring system, teaching method, sophomore students

INTRODUCTION

Structured query language is a vital part of any database language course today. It contains data, view definition and data manipulation statements. It is considered to be a simple and highly structured language. But, students have a lot of difficulties during learning it. In this study we have selected to implement the structured query language part from the database course because students have problem with it the most. This study presents an Intelligent Tutoring System for structured query language called (DB-ITS) that helps students in overcoming these difficulties. DB-ITS is designed as a practice environment; it assumes that the student have previously been introduced to the concepts of database in class. Therefore, the system DB-ITS is not suppose to replace instructors nor the conventional style of education, but to complement it.

This study examines the problems of learning structured query language firstly and then presents the architecture of DB-ITS, examines the system's components and presents an evaluation of the DB-ITS and its effect on performance of students in database.

MATERIALS AND METHODS

In the period between 1/8/2002 and 1/8/2004, 160 students (80 male and 80 female) were randomly chosen from Al-Azhar University in Gaza. All students were 2nd year students enrolling in the database course. The students had no prior knowledge in the subject matter to be taught.

Problems with learning structured query language: Students encounter many problems when learning how to write queries. Some of the problems come from the ordeal of having to memorize database schemas; wrong solutions may contain wrong table or attribute names. Other errors are due to misunderstanding of the components of a query and the relational data model in general. Some of the items students find particularly difficult to understand are grouping, join, aggregate and scalar functions.

Structured query language is usually taught in classrooms, by solving problems on the blackboard, aided by lab exercises. However, students find it hard to learn how to write queries directly by working with a database, because error messages are limited to the syntax only. When the student enters a wrong solution, typically the error message generated by a Database Engine (DBE) will not help much. DBE can only complains about the syntactic errors. Figure 1 shows the type of messages the student may obtain from the system. DB-ITS can generate messages about semantic errors as well; in this case, the student specified two tables in the from clause, when only one of them is really needed as shown in the correct answer.

Overview of DB-ITS: DB-ITS is an intelligent tutoring system that aids students in learning how to write queries. DB-ITS is completely implemented using Delphi 7. The DB-ITS consists of several core models, which are relatively independent of each other to allow easy upgradability and portability to other teaching domains (Anderson *et al*., 1995; Self, 1990; Stern *et al*., 1996):

- A Student Model, which monitors the progress of the student.
- An Expert Model, which stores instructions for each exercise. This model also provides the template by which to compare student actions, to detect constraint violations made by the student.
- An Expert Tutor, which operates on the Student and Expert Models. It is responsible for the validation of each move by the student in an exercise (by means of pattern matching the corresponding steps from both models). The tutor uses the constraint based tutoring module as proposed by Ohlsson (1994) and Blank *et al.* (2004).
- The user interface shell which serves as the bridge or HCI interpreter between students and the DB-ITS (Forgy, 1982; Kim and Hee, 2000; Mayer *et al.*, 2005).

Every user has one copy of the Expert and Student model in the Expert's Knowledge Database. As in the case with other ITSs, DB-ITS tailors instructional sessions to the needs, knowledge, learning abilities and general characteristics of its students. DB-ITS is based on guided discovery, one of the teaching styles commonly found in ITSs. Guided discovery is based on the idea that students should be given opportunities to discover things themselves, rather than being told about them. There are psychological studies (Anderson *et al.*, 1995) which show that students learn better from discovery than from direct instructions and that such knowledge is retained for longer than when learning by being told. Of course, unrestricted exploration is not advisable, especially for novices, as students may waste too much time wandering. The solution is found in providing guidance, in form of hints from the system.

The system is designed as a practice of problem solving environment and as such is not intended to replace human instructions, but to complement it. We assume that students are already familiar with the database theory and fundamentals of structured query language. Students work on their own as much as possible and the system intervenes when the student is stuck or asks for help. In such a way, students maintain a feeling of control. DB-ITS is implemented in Delphi 7. The component of DB-ITS are the user interface, a expert model and a student model, that analyzes student answers. The domain module can solve problems given to students. An ITS must be able to evaluate student answers. DB-ITS does that by comparing student solutions to the correct ones. That is the reason for DB-ITS to require ideal solutions to problems. In order to be able to check the correctness of the student's solution, DB-ITS uses domain knowledge represented in the form of constraints, described in more detail in student model section. The system contains definitions and data of several classical databases (Date, 2003; Elmasri and Navathe, 1994). New databases can easily be added to DB-ITS. DB-ITS also contains a set of problems for classical databases and the ideal solutions for them. At the beginning of a session, DB-ITS selects a problem for the student to work on. When the student enters a solution, the expert model sends it to the student model, which analyzes the solution, identifies mistakes (if any) and updates the student model appropriately. On the basis of the student model, expert model generates an appropriate feedback. When the current problem is solved, or the student requires a new problem to work on, expert model selects an appropriate problem on the basis of the student model.

Student model: A student model develops an understanding of the student's state of mind that can be used to generate instructional actions tailored to the particular student. The task of building a student model is extremely difficult (Zhou, 2000), due to huge search spaces involved and the small amount of information to start from. Several researchers have pointed to the inherent intractability of the task (Ohlsson, 1994; Self, 1990, 1994). If the goal is to model student's knowledge completely and precisely, student modeling is bound to be intractable. However, a student model can be useful although it is not complete and accurate (Ohlsson, 1994; Self, 1994; Stern *et al.*, 1996). Even simple and constrained modeling is sufficient for instruction purposes and this claim is supported by findings that human teachers also use very loose models of their learners and yet are highly effective in what they do (Self, 1994; Beal and Lee, 2005). DB-ITS uses Constraint-based Modeling (CBM) (Ohlsson, 1994) to form models of its students. Student modeling focuses on faults only. Domain knowledge is represented in the form of state constraints, where a constraint defines a set of equivalent problem states. An equivalence class triggers the same instructional action; hence the states in an equivalence class are pedagogically equivalent.

The assumption here is that there can be no correct solution of a problem that traverses a problem state, which violates the fundamental ideas, or concepts of the domain. A violated constraint signals the error, which comes from incomplete and incorrect knowledge.

Student modeling is very fast: in the first step all relevance patterns are matched against the problem state. In the second step, the satisfaction components of constraints whose relevance conditions match the

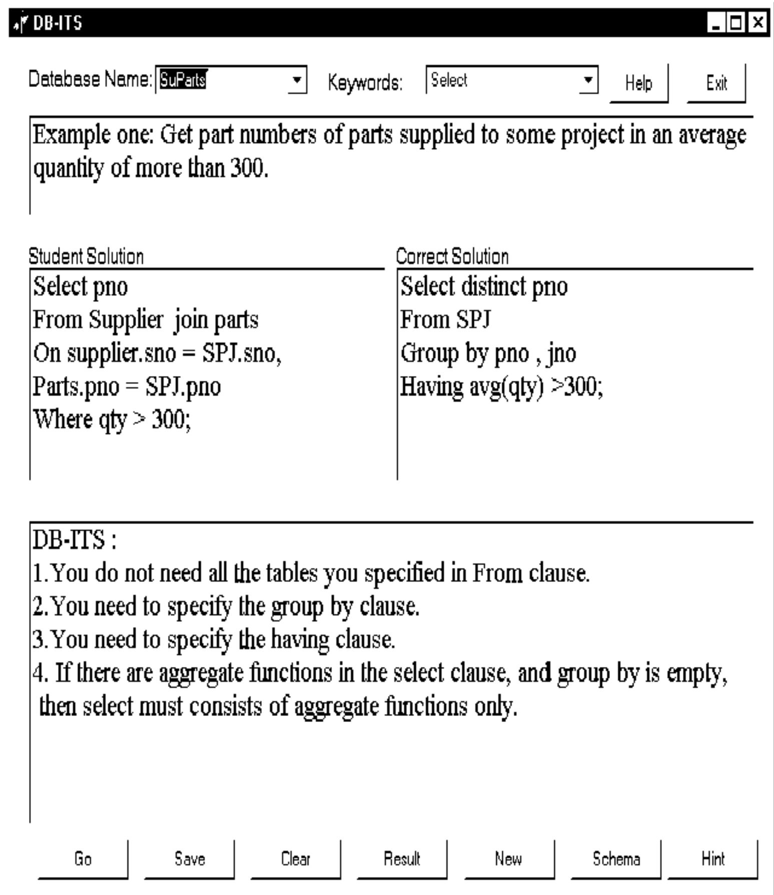

Fig. 1: User interface and feedback from DB-ITS

problem state are matched. If a satisfaction pattern matches the state, the constraint is satisfied and the DB-ITS displays the result of the query to the student. On the other hand, if the constraint is violated. The student model thus consists of all violated constraints. Furthermore, CBM does not require extensive studies of student's bugs and is not sensitive to the radical strategy variability phenomenon (Ohlsson, 1994). The approach is also neutral with respect to the pedagogy, since different pedagogical actions (immediate or delayed ones) may be generated on the basis of the model. DB-ITS models students by looking at the student's solution and by comparing the student's solution to the ideal one. The constraint base of DB-ITS currently consists of 150 constraints, which are acquired by analyzing the domain knowledge (Date, 2003) and from a comparative analysis of correct and incorrect student solutions. Each constraint has a unique number and contains the relevance and satisfaction patterns. Additionally, there is a constraint description and the name of the clause of the select statement the constraint refers to. Relevance and satisfaction patterns can be any logical formulas, consisting of any number of conditions. Some conditions match parts of the student's solution to pre-specified patterns or the ideal solution.

A student model in DB-ITS contains general information about the student, history of previously solved problems and information about the usage of constraints, as demonstrated in the solutions produced by the student. For each constraint, DB-ITS stores information about how many times it was found relevant for ideal solutions, how many times it was actually used by the student and how many times it was used correctly. This information is stored in terms of three indicators (relevant, used and correct), used by expert model for selecting new problems and updated by the student model.

Expert model: Expert module is the heart of the system; it selects problems to be given to students and generates appropriate instructional actions according to the student model. In DB-ITS, instructions can be individualized by generating feedback dynamically and selecting problems.

The level of feedback determines how much information is provided to the student. Currently, there are five levels of feedback in DB-ITS: positive/negative feedback, error flag, hint, partial solution and complete solution. At the lowest level (positive/negative feedback), the message simply informs the student whether the solution is correct or not and in the later case, how many errors there are. An error flag message informs the student about the clause (Date, 2003) in which the error occurred. A hint type message gives more information about the type of error. Here, the student is given a general description of the error. This description is directly taken from the definition of constraint. Partial solution feedback displays the correct content of the clause in question, while the complete solution simply displays the correct solution of the current problem.

It was stated earlier that a student's solution may violate several constraints at the same time (Fig. 1), where 4 constraints were violated simultaneously). In such cases, DB-ITS examines the violated constraints and selects one, which is likely to be a genuine misconception. That is, DB-ITS selects the constraint with the maximum difference between the used and correct indicators. The rationale here is the student has made the same error several times and therefore instruction must start with that constraint. Currently, the student is told the total number of violated constraint, but the error messages only deal with one constraint at the time. This decision is based on our intuition that it would be much easier for the student to deal with errors one at a time.

Problems are also selected on the basis of a student model. DB-ITS examines the student model and selects a problem for constraints that the student is not sure about (i.e., the one with maximal used correctly). In case that there are several massages in various clauses, the expert model will select one of them to start with.

User interface: The interface of DB-ITS is illustrated in Fig. 1. Generally, interfaces for ITSs should be robust, flexible, easy to use and understand. An interface is a mediating device; hence it must provide information about the system itself. At the same time, ITS interfaces are practice of problem solving environments and therefore should be similar to real environments, support the reification of goal structure and reduce the working memory load of students. The interface of DB-ITS reduces the memory load by displaying the database schema and the text of a problem, by providing the basic structure of the query and also by providing explanations of the elements of structured query language. The main window of DB-ITS is divided into three areas, which are always visible to the student. The upper part of the window displays the text of the problem being solved and the student can always remind him/herself easily of the elements requested in the query. The middle part contains the clauses of the select statement, thus visualizing the goal structure. The lowest part is used for displaying: schema of the currently chosen database, feedback with multimedia effects and results when requested. The visualization of a schema is very important; all database users are painfully aware of the constant need to remember table and attribute names and the corresponding semantics as well. DB-ITS users can get the descriptions of databases, tables or attributes, as well as the descriptions of query constructs. The motivation here is to remove from the student some of the cognitive load required for checking the low level syntax and to enable the student to focus on higher level, query definition problems. DB-ITS supports the reification of the goal structure by visualizing the elements (clauses) of query. The student can obtain short descriptions of the roles of various clauses by selecting the appropriate clause or by asking for help from the main menu.

This study aims at achieving the following objectives:

- To develop ITS programs to be used in teaching database.
- To measure the effect of these ITS programs on students' performance in database (measured by exam results).
- To compare the effectiveness of these ITS programs to the conventional teaching methods.
- To measure the effect of sex and multimedia on teaching database.

Questions of this study:

- Are there significant differences in performance between students taught using the conventional method and students taught using DB-ITS programs?
- Are there significant differences in performance between male and female students, in groups taught using DB-ITS?
- Is there a relationship between the use of multimedia effects in DB-ITS and students performance?
- Are there relationships between student evaluation of DB-ITS and their performance in database?

Hypothesis:

- There are no significant differences in performance between students taught using the conventional method and students taught using the DB-ITS programs.
- There are no significant differences in performance between male and female students in the groups taught using DB-ITS programs.
- The use of multimedia effects in DB-ITS has a positive effect on students' performance.
- There is a strong positive relationship between students' evaluation of DB-ITS and their performance in database.

Research frame: The study was conducted on sophomore students enrolled in database course in Al-Azhar University, Gaza.

ITS programs design: ITS programs used in this study are three types:

1. DB-ITS without results of the queries and with multimedia effects. This version of the program focuses on the syntax and semantic of queries that the students respond to questions. Once the query is correct and errors free, the students are informed so.
2. DB-ITS with results of the queries and with multimedia effects. In this version of the program, the program goes one step further in displaying the actual results of the query.
3. DB-ITS without multimedia effects. In this version of the program, the program check for errors and display text messages for the user.

Sample: In the period between 1/8/2002 and 1/8/2004, 160 students (80 male and 80 female) were randomly chosen from Al-Azhar University in Gaza. All students were 2nd year students enrolling in the database course. The students had no prior knowledge in the subject matter to be taught.

Procedure and data collection: The 160 students were selected to participate in this evaluation and randomly assigned to 4 groups (20 male and 20 female for each group):

Group 1 (the control group): This group was taught using the conventional teaching method (lecture and textbook).

Group 2 (the DB-ITS with results group): This group was taught using the DB-ITS with results of queries version of the program and multimedia effects.

Group 3 (the DB-ITS without results group): This group was taught using the DB-ITS without results of queries version of the program and multimedia effects.

Group 4 (the DB-ITS with no multimedia effects group): This group was taught using the DB-ITS with no multimedia effects version of the program.

At the end of instruction, all students were given a test in the subject matter studied and their test grades were recorded. These grades are to be used in examining students' performance level in database exams.

Statistical analysis:

1. A one-way ANOVA test was conducted on the test grades of students in the 4 groups. To determine if there were any significant differences in students performance between the conventional group and the DB-ITS groups. This test was conducted on the entire sample (160 students). If significant differences are to be found, the Scheffe multiple comparison test is to be used to determine differences between group pairs and to investigate possible advantages of one group over another.
2. An independent-sample-t test was conducted on the test grades of male and female students in the DB-ITS groups (2,3,4) one group at a time, to determine if significant differences in performance exist between male and female students in each DB-ITS group.
3. A correlation between students evaluation of DB-ITS groups (2,3,4) and students performance in database was computed to see if there is positive relation between them or not.

RESULTS AND DISCUSSION

Table 1 shows the results of the ANOVA test for the conventional (control) group and DB-ITS groups. The table shows that there are significant differences between the groups. It can be seen from Table 2 that:

- There are significant differences in performance between the conventional group and the DB-ITS with results group in favor of the DB-ITS with results group.

Table 1: ANOVA table for the test grades for groups (1,2,3,4)

Source of variation	Sum of squares	df	Mean square	F	Significance
Between groups	1982.462	3	660.821	25.660	Significant(*)
Within groups	8138.025	316	25.753		
Total	525.597	319			

* Significant at the (0.05) level

Table 2: Scheffe multiple comparisons between group pairs

Group (i)	Group (j)	Mean difference (I-j)	Significance
G1	G2	-6.44(*)	*
	G3	-3.31(*)	*
	G4	-0.93	0.722 (not significant)
G2	G1	6.44(*)	*
	G3	3.13(*)	*
	G4	5.51(*)	*
G3	G1	3.31(*)	*
	G2	-3.13(*)	*
	G4	2.39(*)	*
G4	G1	0.93	0.722 (not significant)
	G2	-5.51(*)	*
	G3	-2.39(*)	*

* The mean difference is significant at the 0.05 level

Table 3: The independent samples t-test results for male and female students test grades DB-ITS groups

Group	Sex	Mean	Std. deviation	t	df	Significance
G2	Male	24.08	6.17	0-1.753	78	0.084 (not significant)
	Female	26.20	4.56			
G3	Male	21.70	4.49	0-.650	78	0.517 (not significant)
	Female	22.33	4.10			
G4	Male	19.43	4.40	0-.343	78	0.733 (not significant)
	Female	19.83	5.92			

Table 4: The correlation between students evaluation of DB-ITS groups and students performance

Group grade	Student evaluation of DB-ITS
G2	0.570**
G3	0.520**
G4	0.433**

** The mean difference is significant at the 0.05 level

- There are significant differences in performance between the conventional group and the DB-ITS without results group in favor of the DB-ITS without results group.
- There are significant differences in performance between the DB-ITS with results group and the DB-ITS without results group in favor of DB-ITS with result group.
- There are significant differences in performance between the DB-ITS with results group and the DB-ITS without multimedia effects group in favor of the DB-ITS with results group.
- There are also significant differences in performance between the DB-ITS without results group and the DB-ITS without multimedia effects group in favor of the DB-ITS without results group.
- The significant differences between groups are ranked as follows: in the first place comes DB-ITS with results, second place comes DB-ITS without results, third place comes DB-ITS without multimedia effects, then the conventional method. However, the removal of multimedia effects from a DB-ITS program takes away much of the advantages over the other DB-ITS programs (with multimedia effects) or the conventional teaching methods.

Table 3 shows the results of the independent samples-t-test between male and female students in the DB-ITS groups. The results show that: there are no significant differences in performance between male and female students in any of the DB-ITS groups. This shows that gender has no effect on students performance when taught using the ITS programs under study. Furthermore, DB-ITS programs are stable and can be implemented effectively.

Table 4 shows the results of the correlation between Students evaluation of DB-ITS groups (2,3,4) and students performance in database. The results show that: there is a strong positive relationship between students evaluation of DB-ITS programs and students performance in database.

CONCLUSIONS

In this study we have introduced an Intelligent Tutoring System for database called DB-ITS. Furthermore, we carried out a preliminary evaluation of this study. The results of this study suggest that DB-ITS can be used as an effective teaching method, however, that does not mean that DB-ITS can replace human instructors. Intelligent Tutoring Systems can be effective only under the supervision of a human instructor. The study also shows that the use of multimedia effects has a strong effect on attracting students attention to the material being taught and thus increasing the level of students learning which in turn gives a better performance in exams.

The removal of multimedia effects seriously decreases students learning level leading to a poor exam performance compared to the conventional method or to other DB-ITS programs.

The study has shown that gender is not a factor affecting the use of the DB-ITS program designs under study, which make these programs more applicable to students. Finally, the results suggest that the performance of students taught using DB-ITS is improved significantly.

REFERENCES

Anderson, J.R., A.T. Corbett, K.R. Koedinger and R. Pelletier, 1995. Cognitive tutors: Lessons learned. J. Learning Sci., 4: 167-207.

Beal, C. and H. Lee, 2005. Creating a pedagogical model that uses student self reports of motivation and mood to adapt ITS instruction. In Workshop on Motivation and Affect in Educational Software, in conjunction with the 12th International Conference on Artificial Intelligence in Education.

Blank, T., L.D. Miller, L.K. Soh and S. Person, 2004. Case-based learning mechanisms to deliver learning materials. In Proc. Intl. Conf. on Machine Learning and Applications (ICMLA'2004), Louisville, KY, December 16-18, pp: 423-428.

Date, C.J., 2003. An Introduction to Database Systems. Addison Wesley System Programming Series: 7th Edition.

Elmasri, R. and S.B. Navathe, 1994. Fundamentals of Database Systems. Redwood: Benjamin/Cummings.

Forgy, C.L., 1982. A fast algorithm for the many pattern/many object pattern matching problem. AI, 19: 17-37.

Kim and J. Hee, 2000. Natural language analysis and generation for tutorial dialogue. Ph.D Thesis, Institute of Technology.

Ohlsson, S., 1994. Constraint-based Student Modeling. Student Modeling: The Key to Individualized Knowledge-based Instruction. Berlin: Springer-Verlag, pp: 167-189.

Self, J.A., 1990. Bypassing the Intractable problem of student modeling. Intelligent tutoring systems: At the Crossroads of Artificial Intelligence and Education. Norwood: Ablex, pp: 107-123.

Self, J., 1994. Formal Approaches to Student Modeling. Student Modeling: The Key to Individualized Knowledge-based Instruction. Berlin: Springer-Verlag, pp: 295-352.

Stern, M., J. Beck and B.P. Woolf, 1996. Adaptation of Problem Presentation and Feedback in an Intelligent Mathematics Tutor. Intelligent Tutoring Systems, New York: Springer-Verlag, pp: 603-613.

Zhou, Y., 2000. Building a new student model to support adaptive tutoring in a natural language dialogue system. Ph.D Thesis, Institute of Technology.