

Diagnosis of the student bugs in the Intelligent Tutoring Systems

تشخيص أخطاء الطالب في نظم التعليم الذكية باستخدام الحاسب

Mohamed Nour*, Nabila Klundeir*, Nadia Hagazi*, Samir Shabcen**

* Electronics Research Institute, Cairo, Egypt

** Faculty of Engineering, Cairo University, Egypt

الملخص

يهدف هذا العمل الى بناء نظام لتعليم مادة الرياضيات تعليم الاعدادى باستخدام الحاسب وذلك عن طريق تطوير احد الأنظمة التعليمية الموجودة (وهو نظام ذكّن للتدريب على الدوائر الكهربائية). والنظام المطور يتجوى على عدة أجزاء منها على سبيل المثال نموذج الخراب وتشخيص أخطاء الطالب وهو الجزء محل الاهتمام هنا. ونموذج الطالب هذا يشمل جميع المعارف والأشياء المتعلقة بسلك الطالب وتعلمة ومتابعة أدائه. وهذا النموذج يستخدم مفاهيم الذكاء الاصطناعي في بناء قاعدة معرفة وبناء آلية استنتاج يمكن استخدامها لى متابعة الطالب وتحليل اجاباته وتحديد أخطائه ومعرفة سبب تلك الأخطاء ومن ثم يتم تحديد مستوى الطالب. وهذا النموذج يقوم بتقييم أداء الطالب، كميًا وكيفيًا. هذا وقد تم تطبيق هذا العمل على حاسب شخصي باستخدام لغة البرولوج في مجال الرياضيات للمرحلة الأعدادية.

Abstract

This paper proposes an intelligent tutoring system for teaching math for the students in the prep schools. The system was not designed from scratch, but it was developed by modifying a certain tutoring system called the intelligent tutoring system for electric circuit exercising. The system contains a set of modules. The concentration here will be on the student model and the students bugs diagnosis. The student model includes all the aspects of the students behavior and knowledge that have repercussions for his performance and learning. The system points out the students bugs. Also, it determines the bugs reasoning (how and why the bugs occurred). The bugs diagnosis is done by using hypothesis generation and hypothesis verifying modules. The system employs the artificial intelligence concepts and methods for monitoring the students performance, tracking errors and offering problem solving advises. Also, the system evaluates the students quantitatively and qualitatively. The system was implemented on the IBM/PC using the prolog language for the math domain.

1. Introduction

The modified intelligent tutoring system (MITS) contains more than one component for making drill and practice in the math domain. The main components are: the problem generation module, the expert module, the student answer checking, the student model, and others. Figure 1 shows the MITS architecture. In MITS, a menu is presented to the student for choosing the question type and the operation. The problem generating module generates the question then, the student answers the question. [1] MITS receives the student answer and interprets it. The student answer checking module checks the format, the type, and the range of the student answer. Then the student answer and the correct system answer (generated by the expert module) will be received by the student model to be compared. If the two answers are matched, this means that the student answer is correct. Then the evaluation module begins its work. If the two answers are mismatched this means that there is a certain bug in the student answer. In this case the diagnosis module begins its work and points out the bug, how and why the bugs occurred. In all cases, the evaluation module maintains a student profile for storing the student performance, such as the right answers, the wrong answers, the number of trials, the bugs types, and all the student responses. The concentration in this work will be on the student model specifically the diagnosis process. The diagnosis process is the process of forming and updating the student model by analyzing data made available to the system. The diagnosis process usually requires search, whether the system is trying to reconstruct a goal structure to interpret the student behavior. In general, the diagnosis process is a hard artificial intelligence problem which involves the formulation and evaluation of competing hypothesis. The organization of this paper will be as follows: section 2

Accepted February, 16, 1997

presents the student model and the proposed algorithm for the diagnosis process. Section 3 presents the student evaluation, while section 4 presents the conclusion remarks followed by a set of references.

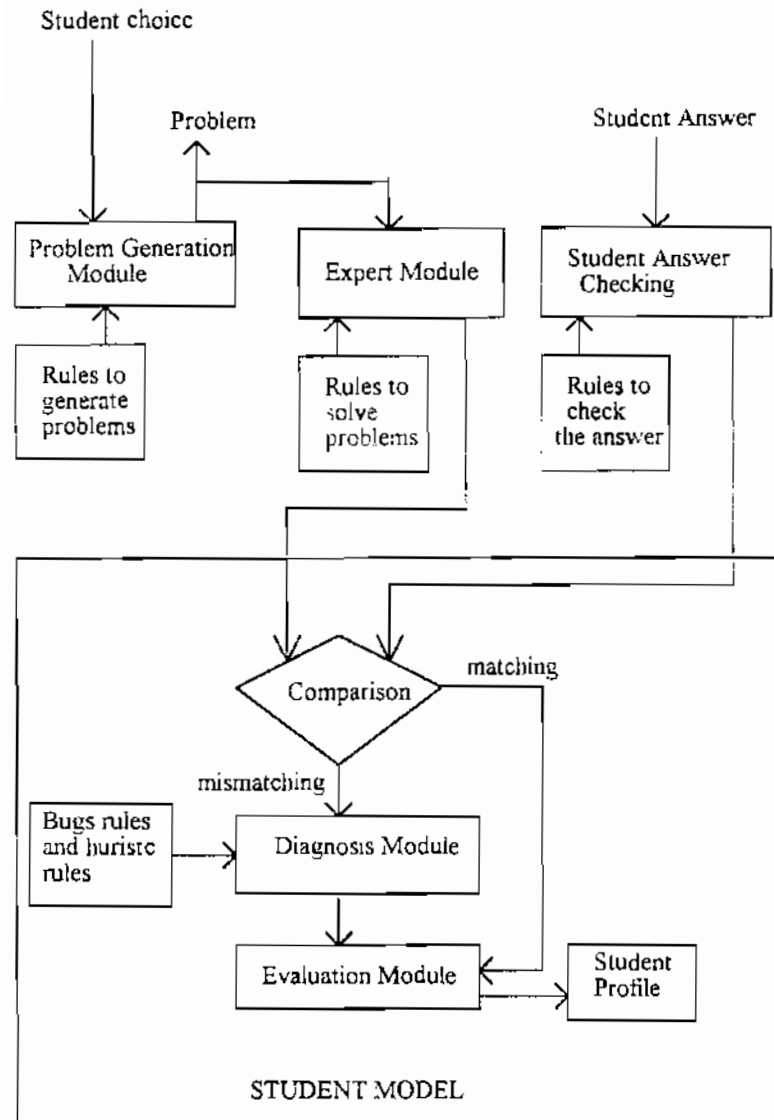


Figure 1: The modified intelligent tutoring system (MITS) architecture.

2. The student model for the modified intelligent tutoring system

The student model maintains the student performance. The student model contains two main modules: the diagnosis module and the evaluation module[2,3]. The diagnosis module points out the student errors. Also, it classifies the error type and its

reasoning. The diagnosis process is done by supporting the student model by hypothesis generation and verification modules as shown in the following sections

The implemented algorithm for the diagnosis process module

The diagnosis process begins by comparing the expert answer and the student answer. The format, type, and range of the student answer will be checked by the student answer checking. On the other hand, the comparison process covers different questions types such as True/False questions, Multiple choice questions, Essay questions, and others. The 'set' in Math was selected as a workload. [4] Figure 2 shows the selected different operations can be done on the sets. Some of the questions types exist in the Math subject are as follows.

Q1. Mark T for true sentences or F for false sentences for the following;

$$2 \in \{2,7,9\}$$

$$\{2,5,9\} \subset \{4,7,9\}$$

Q2. Assume $X = \{2\}$, $Y = \{5,7,11,13\}$, $Z = \{5,7,11,13,9\}$

Write the number of elements in each of the following;

1. $n(X \cap Y)$

2. $n(X \cup Z)$

3. $n(X \cup Y \cup Z)$

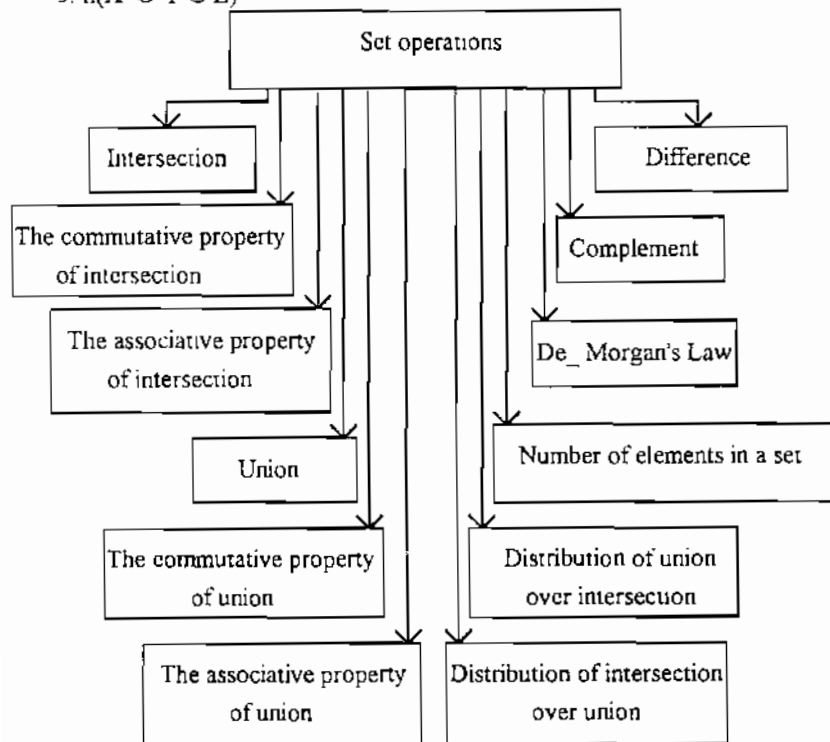


Figure 2: The set operations

Q3. Assume $X = \{3,5\}$, $Y = \{5,7,11\}$, $Z = \{5,17,11,1,3\}$

Write the number of elements in each of the following;

1. $X \cup (Y \cup Z)$
2. $X \cup (Y \cap Z)$

The comparison process differs according to the question type. W.r.t the first question type (True/False questions), the comparison will be done between characters. W.r.t the third questions type, the comparison will be done between two sets (the expert answer and the student answer). In this case the comparison will be done as follows: compare the number of elements of the two sets. If the two answers did not match, this means that the two sets are not equal. If they are matched, then compare the elements values of the two sets. If the elements values of the two sets did not match, this means that the two sets are not equal; but if they matched this means equality of the two sets. The system transfers control to the evaluation module if the two answers matched. If the two answers mismatched, the system is going to check the student answer w.r.t the domain bugs. In this case, there are two types of bugs: repeating elements bugs and strange elements bugs. If the student answer contains repeating elements, this means; the student has misunderstanding of the main concepts. Also the student answer must be checked to know if there is an element(s) in his answer not exist in the system answer. If the student answer contains a domain bug, the system gives the student an advice. Also, the student will be givee another chance for answering the question. If the student answer is still wrong, the system determines the student bugs reasoning. The reasoning of bugs can be determined by using two modules. The hypothesis generating module, and the hypothesis verifying module. The hypothesis generating module generates many possible bugs. The bugs generation depends on the question nature. i.e. there are two types of questions operations: One operation questions, and multiple operations questions. For the one operation questions (as an example), the hypothesis generation module generates the bugs as shown in figure 3.

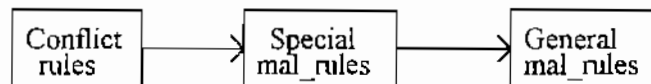


Figure 3. Bugs used by the hypothesis generation module

[5,6] The hypothesis generating module accepts the correct rule number from the expert module. Then it recognizes the conflict rules by using the database containing the integer numbers indicating the conflict rules associated with each rule. It chooses one of these conflict rules and knows if it can give the wrong answer or not by using heuristic rules. If yes, it passes the conflict rule to the verification module. If not, it chooses another one and repeats the last. The verification module knows if the hypothesis rule gives a matching answer or not. Matching means the hypothesis is correct (the generated bug is the actual bug). On the other hand, mismatching means the hypothesis is incorrect. The hypothesis generating module will generate other hypothesis and all the last are repeated. If all the available conflict rules are finished without matching, the hypothesis generation module generates another bugs type called the special mal rules. Choosing of one of the special mal_rules will be selected for the given correct rule and then it will be checked by using domain dependent heuristic rules to know if it can give the student answer or not. If yes, the verification module knows if the hypothesis rule gives a matched answer or not, while matching

between the two answers means the hypothesis is correct. If all the special mal_rules are finished without matching, the hypothesis generation module generates another type of bugs called the general mal_rule. The hypothesis generating module chooses one of the general mal_rule which affects the final answer, checks it by using the domain dependent heuristic rules. This is necessary to know if this can give the student answer or not. The verification module knows if this hypothesis gives a matched answer with the student answer or not [7,8]. If all possible hypothesis bugs are finished without matching with the student answer, the system explains the operation concerned with that rule and gives the correct answer. For the procedure questions, the hypothesis generation module generates the bugs in the sequence shown in figure 4. The sequence of bugs generation is : expression expansion bugs, wrong order bugs, and missing operation bugs.

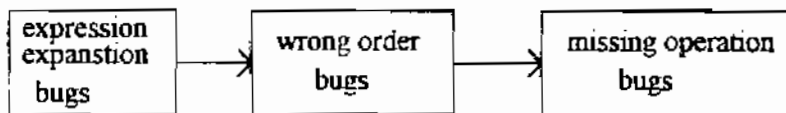


Figure 4. The bugs generation for procedure questions

The expression expansion bugs in figure 4 means, if the expression has expansion, the hypothesis module generates some of the general expressions. Ex. for expression $(X \cup Y)$, the right answer is $\overline{X \cap Y}$. One of the famous wrong expansion is $\overline{X} \cup \overline{Y}$ without changing the operator. So, the hypothesis module generates that bug by changing the operator. Also, the student answer may contain mistakes due to the wrong order evaluation. The hypothesis generating module can recognize that mistake by comparing the student answer with the common bugs due to the wrong order. Also the student mistakes may be done due to "missing operation". The following example illustrates that concept.

$X = \{1,2,3,6\}$, $Y = \{2,5,7\}$, $Z = \{1,3,11,7\}$

List the elements of the expression $X \cup (Y \cap Z)$.

The student answer is $\{7\}$. The diagnosis process begins by comparing the student answer $\{7\}$ with the expert answer $\{1,2,3,6,7\}$. The two answers are not matched, so the student answer has a bug. In this case the hypothesis module recognizes the reasoning of the student bug. The system proposes that the student answer may be obtained by evaluating the expression $(X \cup Y) \cap Z$. In this case, the answer is $\{1,3,7\}$ which is not matched with the student answer. The hypothesis module ignores that possibility (wrong order bug). It proposes another bug, i.e. the student answer is obtained by evaluating $(Y \cap Z)$ without taking X and the union operation into consideration. In this case the system answer is $\{7\}$ which is matched with the student answer, i.e. the error occurred due to missing operation. So, the diagnosis process sends a message to the student "Sorry, incorrect answer" The diagnosis process confirms that concept through a dialogue between the system and the student as follows:

System: $X = \{1,2,3,6\}$, $Y = \{2,5,7\}$, $Z = \{1,3,11,7\}$

List the elements of the expression $X \cup (Y \cap Z)$.

Student: $\{7\}$

System: Sorry, Please answer the following questions

$X = \{1,2,3,6\}$ $Y = \{2,5,7\}$ $Z = \{1,3,11,7\}$

$X \cup (Y \cap Z)$
 " Which operation will applied first ?"
 "1- union 2- intersection"
 student : 2
 System : " correct answer"
 System : " $Y \cap Z =$ "
 Student: {2, 7}
 System : Wrong answer, because 2 is not a common element. The right
 answer is {7}
 System: $X \cup (Y \cap Z)$
 $\{1,2,3,6\} \cup \{7\} =$
 Student {1,2,3,6,7}
 System : "Bravo, correct answer. "

The system uses always the decomposition concept for the long questions. Figure 5 shows the global flow chart for the diagnosis process.

3. The student profile and the evaluation module

In the modified intelligent tutoring system (MITS), the student enters the answer of the displayed questions. Then MITS is going to check the correctness of the student answer. It recognizes the student bugs, the bugs location, and the bugs reasoning in the incorrect answers. This is done by the evaluation process[9,10]. In fact, MITS maintains two types of evaluation: the qualitative evaluation and the quantitative evaluation. The quantitative evaluation (i.e. grading or scoring) is based on the number of correct answers w.r.t. the total number of answers. On the other hand, the qualitative evaluation concentrates on the different types of the operations used, and the understanding level of each operation. MITS begins to ask the student about his name, so the system knows if the student is an old student or a new one. MITS creates a student profile if the student is new and opens the old profile if the student is not new.

The student profile contains a set of arguments such as the question type, the operator used, and the answer type. On the other hand, the displayed questions are classified into groups. The student profile is updated after answering any question. The evaluation process is based on the information stored in the student profile. For the quantitative evaluation, the evaluation process counts (for each question type) the number of correct answers and the number of wrong answers. Also, it calculates the student score. $\text{Score} = \text{number of correct answers} / \text{number of asked questions}$. W.r.t. the qualitative evaluation, the evaluation module keeps track of the student response status for each operator. It monitors the student answer for each operator in all questions. i.e. the evaluation module monitors the student progress level through the student performance.

By using the student profile, MITS controls the types of displayed questions. Also MITS can switch from the questioning mode to the explanation mode to redisplay the weak point concepts to the student. MITS also gives a full report of the student performance supported by graphics. Figure 6 shows the understanding level of the intersection operation (in sets) exist in different seven questions. The figure shows a good performance for a real student toward the correct way.

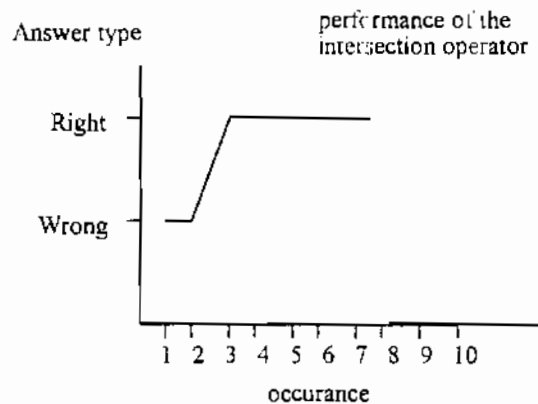


Figure 6: Student progress for intersection operation

Finally, MITS provides the student profile to the student indicating his performance. i.e. the weak points, the rate of grasping, and the number of iterations for each answer will be displayed. MITS gives a full report about the misunderstanding concepts and the understanding concepts to the student. Also, MITS presents advises to the student according to his understanding and grasping level.

4. Conclusions

The modified intelligent tutoring system (MITS) structure was presented. The concentration was on the diagnosis process, which was done on the IBM/PC using prolog language for math domain (sets). MITS presents a large number of random questions by the problem generation module. MITS receives the student answer, analyzes it, and compares it with the correct system answer generated by the expert module. The proposed diagnosis algorithm can point out the student bugs. Also, it

recognizes the bugs location and the bugs reasoning. The student bugs may take place due to misunderstanding in the domain concepts. i.e. the student may use some rules in a conflict way. Also, the student bugs may take place due to mistakes in expressions expansion. The student may apply a certain rule before the other rules in a wrong way (i.e. wrong order bugs). Also, the student in some cases may forget handling some rules (missing operation bugs). The hypothesis generation and the verifying generation activities in the diagnosis module can detect and predict all the previous student bugs by using a set of mal_rules. Those mal_rules are conflict rules, special mal_rules, and general mal_rules. The conflict rules recognize the student conflict concepts. The special mal_rules are based on the math domain characterization (i.e. domain dependent) while the general mal_rules are domain independent. MITS monitors the student performance. Also, it evaluates the student progress quantitatively and qualitatively. MITS uses a student profile to register all the student behavior. MITS presents a full report indicating the student score, the number of wrong answers, the number of correct answers, and the number of iterations for answering each question. MITS also recognizes the student weak point concepts in the selected domain, and the student progress in each concept.

5. References

- [1] A.Yoshikawa & et.al, "Intelligent Tutoring System for Electrical Circuit Exercising", IEEE Transactions on Education, Vol. 35 No. 3, pp. 222-225, August 1992.
- [2] William D.Siuru," Challenger: A domain-independent intelligent tutoring system", IEEE Transactions on Education, Summer 1989.
- [3] Sleeman, D.H."PIXIE - a shell for developing intelligent tutoring systems.", Ablex publishing, Norwood, New Jersey, 1987
- [4] Ibrahim El-Dessouky& et al, "Mathematics for First Year Preparatory Schools", The Egyptian International Publishing Co.-Longman, 1988.
- [5] Etienne Wenger & et.al, "Artificial Intelligence and Tutoring Systems", Morgan kaufman publishers, Inc., 1987.
- [6] Hugh Burns & et.al."Intelligent Tutoring Systems-Evolutions in Design", Lawrence Erlbaum Associates publishers, 1991
- [7] Morris W. Firebaugh, "Artificial Intelligence: A Knowledge-based approach", PWS-KENT Publishing Company, 1990.
- [8] Ron Zambo& et.al. "A professional Development Model for Technology Mathematics and Science", The Proceedings of the Seventh International Conference of Information Technology and Education (SITE96), PP.192-195 Phoenix, Arizona, USA, 1996.
- [9] Peter Zorkoczy, "Educational Scenarios for Telecommunication Applications", Collaborative Dialogue Technologies in Distance Learning, by M. Felisa Verdejo, and Stefano A. Cerri, Springer-Verlag Berlin Heidelberg, PP.210-218, 1993
- [10] Mohamed Nour and Mohamed Younis," Tutee: An Intelligent Tutoring System in Math", The Proceedings of the Seventh International Conference of Information Technology and Education (SITE96), Phoenix, Arizona, USA, 1996