

Adaptive Intelligent Tutoring System for Learning Computer Theory

MOHAMMED A. M. AL-NAKHAL

Department of Information Technology
Faculty of Engineering & Information Technology
Al-Azhar University, Gaza, Palestine

SAMY S. ABU NASER¹

Professor of Artificial Intelligence
Department of Information Technology
Faculty of Engineering & Information Technology
Al-Azhar University, Gaza, Palestine

Abstract:

In this paper, we present an intelligent tutoring system developed to help students in learning Computer Theory. The Intelligent tutoring system was built using ITSB authoring tool. The system helps students to learn finite automata, pushdown automata, Turing machines and examines the relationship between these automata and formal languages, deterministic and nondeterministic machines, regular expressions, context free grammars, undecidability, and complexity. During the process the intelligent tutoring system gives assistance and feedback of many types in an intelligent manner according to the behavior of the student. An evaluation of the intelligent tutoring system has revealed reasonably acceptable results in terms of its usability and learning abilities are concerned.

Key words: Intelligent Tutoring System, Authoring Tool, ITSB, computer theory

¹ Corresponding Author: abunaser@alazhar.edu.ps

INTRODUCTION

In mathematics and computer science, computer theory course is the branch that involved with how efficient a problem may be solved using algorithmic computational model. The field consists of three main divisions: theory of computability, language and automata theory, theory of computational complexity that is associated with the question: "What are the essential capabilities and restrictions of computers?" To be able to implement a rigorous study of computation, computer experts who work with a computers mathematical abstraction named computational model. There are a few models that are being used, but the most frequently studied is the Turing machine. Computer experts study the Turing machine due to its simplicity in formulation, may be examined and used to verify results and since it denotes what numerous experts debate the most influential expected model of computation). It may appear that the hypothetically infinite memory is an unattainable attribute, but a decidable problem resolved using Turing machine will need solely finite a quantity of memory. Thus, in theory, any problem that can be resolved by a Turing machine can be resolved by a computer that incorporates finite a quantity of memory [2,3].

The basic theory of learning styles is that different people learn in a different ways. One way to see the learning styles is to connect them with the learning cycle advocated by Kolb, where learning is seen as a continuous process based on practical experience that incorporates a set of observations and reflections [6].

Later, Mumford Honey developed this model by creating a questionnaire for the styles of learning using the model suggested by Kolb. Authors identified four learning styles, related to the different four stages of the learning cycle which was suggested by Kolb. They are theorist, activist, pragmatist, and reflector.

These styles became very important in education, given the change in the educational prototype presented by the changes to the society of knowledge. The ultimate learning prototype introduced new learning settings, which are progressively more heterogeneous, where it is significant to be taken into consideration the learning styles of every student to offer an education more actual and concentrated on the student [16,18].

Figueiredo and Afonso considered the setting and content as the main parts of the learning model. The learning model identified the learning activities as the settings in which people learn. The content is the data that is organized and contains: text, resources, multimedia and lessons. The setting is a group of surroundings that are appropriate to the user to construct knowledge by its linking to the content [18].

In this model, the instructor has a mutual role in the arrangement of content and generating the learning setting. The setting can be a lecture hall or a virtual learning site, in which the role of instructor is more concentrated on content in the case of a lecture hall, and the setting in the case of a virtual learning site.

Contents undertake the role of broadcast the knowledge, where data is converted into knowledge by a known learning activity [14,24].

The incorporation of intelligent systems in the learning manner support permits an adaptation of content and settings into the learning style of every student, making adaptive tools to support cooperation [24]

The intelligent tutoring system for learning computer theory was designed and developed using ITSB authoring tool [1].

LITERATURE REVIEW

There are many intelligent tutoring systems designed and developed for the education purposes. Some of these ITS

dedicated to teaching computer science like [11, 12, 13, 15], English language such as [19] and Mathematics such as [22]. Teaching Java objects Programming language[5], ITS for helping Computer Science students to learn debugging skills [11], ITS helping Computer Science students to learn Computer Programming [8, 23], an Intelligent Tutoring System for Teaching Grammar English Tenses[9], ITS for helping students who are 8-to12-year-olds to learn Primary Mathematics[22], A comparative study between Animated Intelligent Tutoring Systems (AITS) and Video-based Intelligent Tutoring Systems (VITS) [4], An agent based ITS for Parameter Passing In Java Programming[12], Java Expression Evaluation [18], Linear Programming[13,17], Design and Development of Diabetes Intelligent Tutoring System [16], effectiveness of e-learning[7], computer aided instruction[21], effectiveness of the CPP-Tutor[10], teaching AI searching algorithms[24], teaching database to sophomore students [15], Predicting learners performance using NT and ITS [20], and intelligent tutoring system for teaching advanced topics in information security[26], development and evaluation of the Oracle Intelligent Tutoring System (OITS)[27].

ITS ARCHITECTURE

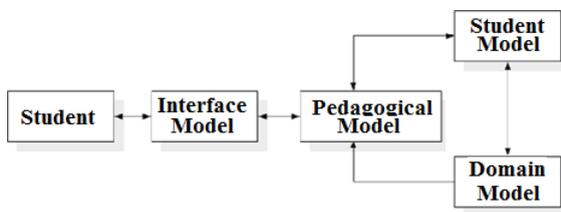


Figure 1: Computer Theory ITS Architecture

A general intelligent tutoring system architectures as in Figure 1 consists of the student model, interface model, domain model, and pedagogical model [8].

In the computer theory intelligent tutoring system, the student plays the main role. All the characteristics of the

system have the task to adapt the pedagogical, material, the interface to the student profile and his preferring.

The domain model is the knowledge management system that stores the material which will be presented to the student.

The student model signifies learning style, the student behavior, motivation level, his profile, and his preferences. This model is grounded on artificial intelligence skills that emulate the human behavior. All the student behavior is stored in the system and used for reasoning and adapting the domain model to the student needs.

The pedagogical model acts as virtual instructor, exhibiting the materials in a suitable sequence, according to learning style and student skills. This process is an interactive and this model job is to illustrate the concepts to the student and supporting all the learning process [9].

The interface model has a very significant job. When an ITS have strong pedagogical, domain, and student models, but the interface model is very bad, the ITS will not be useful because the interface is the door of all the system and has the capability to lid all the attention of the student. To develop a good interface model, it is necessary to take into consideration the usability matters of a user computer interface, since this model communicates with the user and the other models of the system [1, 8].

DOMAIN MODEL ARCHITECTURE

The topics covered in this Intelligent Tutoring system as follows [2,3]:

- Basic concepts and definitions; Set operations; partition of a set
- Equivalence relations; Properties on relation on set.
- Proving Equivalences about Sets.
- Central concepts of Automata Theory.

- Regular Expressions; Operations on Regular expressions
- Finite Automata and Regular Expressions.
- Conversion from FA and regular expressions.
- Deterministic Finite Automata (DFA); Minimization of DFA.
- Non-Deterministic Finite Automata (NFA).
- Equivalence of Deterministic and Non-Deterministic Finite Automata.
- Equivalence between DFA, NFA, NFA- Λ
- Context-Free Grammars.
- Parse Trees; Ambiguity in Grammars and Languages.
- Standard Forms; Chomsky Normal Forms;
- Minimization of CFG's.
- Pushdown Automata (PDA).
- Deterministic and Non-Deterministic (PDA); Formal definition of NPDA.
- Transition functions of NPDA; NPDA Execution.
- Accepting Strings with NPDA; Equivalence of PDAs and CFG.
- The Turing Machine.
- Programming Techniques for Turing Machines; Formal definition of TM's.
- Complexity issues and analysis

STUDENT MODEL ARCHITECTURE

The intelligent tutoring system logs the student data in a database. It stores the basic data of the students such his/her name, user id, current score, overall score, lesson exercises done, lesson exercises difficulty level reached, and number questions of a lesson exercises tried. These data is stored for later usage in other models such pedagogical model.

PEDAGOGICAL MODEL ARCHITECTURE

The pedagogical model analyze the stored information about the student to decide the customized feedback about the student educational level, when the student solve an exercise correctly or incorrectly, when the student score in a lesson exercises is bad, average or good. For example, when the student's current score in a particular lesson exercises is less than %50:

1. The pedagogical model display a notification message telling the student "your performance in this lesson exercises is bad and I suggest that you go back to lesson and study it well then come back to the exercises and try again",
2. The pedagogical model close the exercises screen and take the student back to the lesson screen so the user study the material of the lesson again

But when the student's current score in a particular lesson exercises is between %50 and less than %70:

1. The pedagogical model display a notification message telling the student "your performance in this lesson exercises is not good enough" and show the score in the notification message.
2. The pedagogical model collect the questions related to the same lesson and re-arrange then randomly and preset them to the student to solve.

However, when the student's current score in a particular lesson exercises is greater than %70:

1. The pedagogical model display a notification message telling the student "your performance in this lesson exercises is good and you passed this lesson is successfully" and the score in the notification message is shown to the student.

2. The pedagogical model takes the student to the next difficulty level of the lesson to solve.

USER INTERFACE MODEL

In this student interface, student chooses the desired lesson to learn.

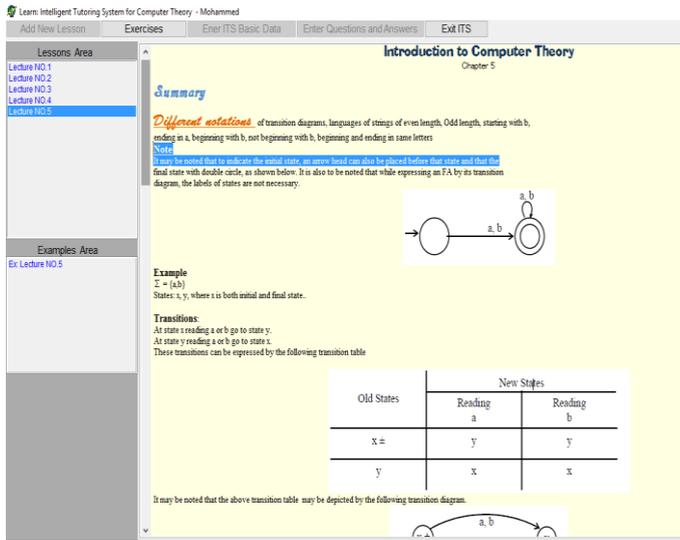


Fig. 2: Intelligent Tutoring System Lesson area

In this student interface, student chooses the lesson exercises and tries to solve as in Fig 3. After the student solves the question, he can check to see if his answer is correct or not. If the student cannot solve the question, he can see the solution of the question. The student can check his current score and overall score anytime (See Fig. 4). In case the student does not get at least 50% as a current score and the end of the lesson exercises he get the notification message as seen in Fig. 5.

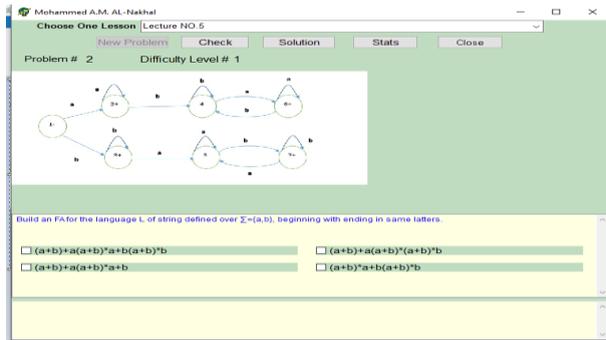


Fig 3. Intelligent Tutoring System Exercises area

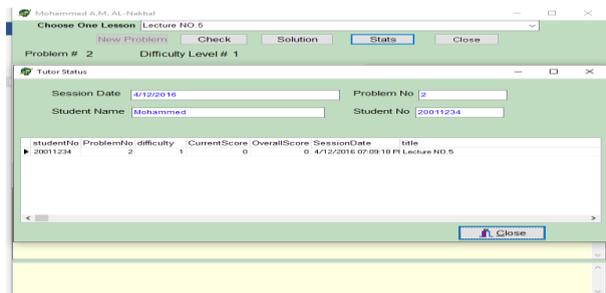


Fig. 4: Intelligent Tutoring System display student statistics

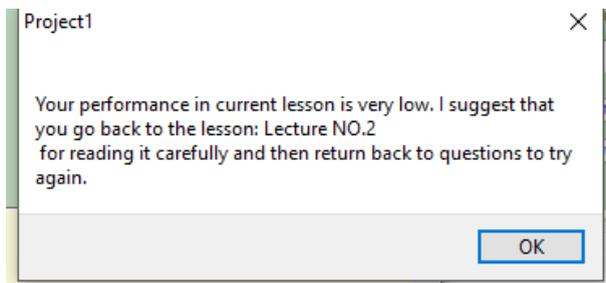


Fig. 5: Intelligent Tutoring System display a notification for the student

EVALUATION

The intelligent tutoring system was presented to some specialists in the field and was applauded by some, while others requested to improve the system by increasing the lessons, questions, and exercises. But on the educational material side,

they stated that it is easy for students to understand the lessons and follow-up at any time they want.

CONCLUSION

An Intelligent Tutoring System for teaching students computer theory was designed and developed using the ITSB authoring tool. We have tried to keep the design as simple as possible in terms of explaining the material instruction, while ensuring that the student take advantage of the questions and find solutions for those who could not answer. An evaluation was carried out to see the student and teacher satisfaction of the Intelligent Tutoring System. The results of the evaluation were satisfactory.

In the future, we will increase the lessons to cover a greater range of educational material and increase number of exercises of each lesson.

REFERENCES

1. Abu Naser, S. S. (2016). ITSB: An Intelligent Tutoring System Authoring Tool. *Journal of Scientific and Engineering Research*, 3(5), 63-71.
2. <http://www.intechopen.com/books/advances-in-learning-processes/advances-in-artificial-intelligence-to-model-student-centred-vles>.
3. Daniel I. A. Cohen. (1997). *Introduction to Computer Theory*, Prentice-Hall, Second Edition, 1997.
4. Abu Naser, S.S. (2001). A comparative study between Animated Intelligent Tutoring Systems (AITS) and Video-based Intelligent Tutoring Systems (VITS), *Al-Aqsa University Journal*,5(1) Part,1.
5. Abu-Naser, S., Ahmed, A., Al-Masri, N., Deeb, A., Moshtaha, E., & AbuLamdy, M. (2011). An Intelligent

- Tutoring System for Learning Java Objects, International Journal of Artificial Intelligence and Applications (IJAIA), 2(2).
6. Lesgold, A., Katz, S., Greenberg, L., Hughes, E., & Eggen, G. (1992). Extensions of Intelligent Tutoring Paradigms to Support Collaborative Learning. In S. Dijkstra, H. Krammer, & J. van Merriënboer (Eds.), *Instructional Models in Computer-Based Learning Environments*. Berlin: Springer-Verlag, 291-311.
 7. Abu-Naser, S., Al-Masri, A., Abu Sultan, Y., & Zaqout, I. (2011). A prototype decision support system for optimizing the effectiveness of e-learning in educational institutions, *International Journal of Data Mining & Knowledge Management Process (IJDMP)*, 1, 1-13.
 8. C. J. Butz, S. Hua, R. B. Maguire, "A web-based Bayesian intelligent tutoring system for computer programming", *Web Intelligence and Agent Systems*, Vol.4, No.1, pp.77-97 · January 2006.
 9. Alhabbash, M. I., Mahdi, A. O., & Abu Naser, S.S. (2016). An Intelligent Tutoring System for Teaching Grammar English Tenses, *European Academic Research*, 4(9).
 10. Naser, S. (2009). Evaluating the effectiveness of the CPP-Tutor an intelligent tutoring system for students learning to program in C++, *Journal of Applied Sciences Research*, 5(1), 109-114, <http://www.aensiweb.com/JASR/>.
 11. Carter, Elizabeth and Blank, Glenn D, "An Intelligent Tutoring System to Teach Debugging", *Artificial Intelligence in Education: 16th International Conference, AIED 2013, Memphis, TN, USA, July 9-13, 2013*.
 12. Naser, S. A. (2008). An Agent Based Intelligent Tutoring System For Parameter Passing In Java Programming, *Journal of Theoretical & Applied Information Technology*, 4,7.
 13. Naser, S. A., Ahmed, A., Al-Masri, N., & Abu Sultan, Y. (2011). Human Computer Interaction Design of the LP-ITS:

- Linear Programming Intelligent Tutoring Systems, International Journal of Artificial Intelligence & Applications (IJAIA), 2(3),60-70.
14. Figueiredo, A. D., & Afonso, A. P. (2005). Context and learning: A philosophical framework. In A. D. Figueiredo & A. P. Afonso (Eds.), *Managing learning in virtual settings: The role of context* (pp. 1-22). Hershey, PA: Information Science.
 15. Naser, S. S. A. (2006). Intelligent tutoring system for teaching database to sophomore students in Gaza and its effect on their performance, *Information Technology Journal, Scialert*,5(5),916-922.
 16. Almurshidi, S. H., & Abu Naser, S. S. (2016). Design and Development of Diabetes Intelligent Tutoring System, *European Academic Research*, 4(9).
 17. Naser, S. S. A. (2012). A Qualitative Study of LP-ITS: Linear Programming Intelligent Tutoring System, *International Journal of Computer Science & Information Technology*, 4(1),209-220, Academy & Industry Research Collaboration Center (AIRCC)
 18. Abu Naser, S. (2008). JEE-Tutor: An Intelligent Tutoring System for Java Expression Evaluation, *Information Technology Journal, Scialert* , 7(3),528-532.
 19. Mona H. Mahmoud, Sanaa H. Abo El-Hamayed, "An Intelligent Tutoring System for Teaching the Grammar of the Arabic Language, *Journal of Electrical Systems and Information Technology*", 2016, Available: <http://dx.doi.org/10.1016/j.jesit.2016.04.001>.
 20. Naser, S. S. A. (2012). Predicting learners performance using artificial neural networks in linear programming intelligent tutoring system, *International Journal of Artificial Intelligence & Applications*, 3,2, 65-73, 2012,Academy & Industry Research Collaboration Center (AIRCC).

21. Naser, S.S A., & Sulisel, O. (2000). The effect of using computer aided instruction on performance of 10th grade biology in Gaza.
22. Hennessy, Sara, Tim O'Shea, Rick Evertsz, and Ann Floyd. "An Intelligent Tutoring System Approach to Teaching Primary Mathematics." *Educational Studies in Mathematics* 20, no. 3 (1989): 273-92. <http://www.jstor.org/stable/3482472>.
23. Naser, S.S. A. (2008). Developing an intelligent tutoring system for students learning to program in C++, *Information Technology Journal, Scialert*,7(7),1055-1060.
24. Naser, S.S.A. (2008). Developing visualization tool for teaching AI searching algorithms, *Information Technology Journal, Scialert*,7(2), 350-355.
25. Goodman, B., Hitzeman, J., Linton, F., & Ross, H. (2003). Towards Intelligent Agents for Collaborative Learning: Recognizing the Role of Dialogue Participants. In *Proceedings of the International Conference on AI in Education*.
26. Mahdi, A. O., Alhabbash, M. I., & Abu Naser, S. S. (2016). An intelligent tutoring system for teaching advanced topics in information security, *WWJMRD*, 2(12), 1-9.
27. ALDahdooh, R., & Abu Naser, S.S. (2017). Development and Evaluation of the Oracle Intelligent Tutoring System (OITS), *European Academic Research*, 4(10).
28. Alnajjar, M., & Naser S.S.A. (2015). Improving Quality Of Feedback Mechanism In Un By Using Data Mining Techniques, *International Journal of Soft Computing, Mathematics and Control*, 4(2).
29. Abu Naser, S. (1993). A methodology for expert systems testing and debugging, North Dakota State University, USA 1, 1-130
30. Naser, S.S.A. (1999). Big O Notation for Measuring Expert Systems complexity, *Islamic University Journal - Gaza* 7 (1), 77-57.