

ADO-Tutor: Intelligent Tutoring System for leaning ADO.NET

IBRAHIM A. EL HADDAD

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:

This paper describes an Intelligent Tutoring System for helping users with ADO.NET called ADO-Tutor. The Intelligent Tutoring System was designed and developed using (ITSB) authoring tool for building intelligent educational systems. The user learns through the intelligent tutoring system ADO.NET, the technology used by Microsoft.NET to connect to databases. The material includes lessons, examples, and questions. Through the feedback provided by the intelligent tutoring system, the user's understanding of the material is assessed, and accordingly can be guided to different difficulty level of exercises and/or the lessons. The Intelligent Tutoring System was evaluated by a group of users and the results were more than satisfactory in terms of the quality of the material and the design of the system.

Key words: Intelligent Tutoring System, Authoring Tool, ITSB, Microsoft, .Net Frameworks, database, ADO.NET, SQL Server

¹ Corresponding author: abunaser@alazhar.edu.ps

INTRODUCTION

The beginning of ASP back in late 1996 when ADO 1.0 was released representing a new way of dynamically reclaiming data from a database. Though ADO was at its early childhood then, and was something of an offshoot of DAO and RDO, nonetheless it represented a new bold direction. Each subsequent version of the technology takes us one step closer to ADO 2.6, when development finally came out. At this point, evolving from the shadows came out the most radical framework to date – .NET, with it the very powerful and complete ADO.NET [1,2].

This new data module, presented with .NET, offered an exciting new method to data access. However the methods, and logic used to connect to databases with ADO.NET weren't shockingly different from those used with its ancestor, ADO.NET had a lot to bargain. What was sole about this technology was the construction underneath it all, its powerful methodology to data organization, and the elasticity in the next level of data-presenting devices [3].

ADO, for its time, was definitely successful. Even though the re-releases of the technology had not indicated intense change for ADO, it attained what it had to within its own construction. Though, it was repeatedly overwhelmed by problems that fenced its management of disengaged data stores, and functioning appropriately and tersely with XML. This is where ADO.NET came in – these are the two things the new technology simply succeeds, as XML is the essential constituent of the whole .NET Framework! ADO.NET was simply produced to solve all the lacks found in ADO, and offers developers with the power to attain more with less [4-6].

ADO-Tutor was designed and developed using ITSB authoring tool [23].

The goal of our intelligent tutoring system was to help users learn easily how to use ADO.NET and to connect to databases.

The benefits of Intelligent Tutoring Systems include[7]:

- ITS are available at any time of the day, even late at night before an exam.
- ITS provide real-time data to teachers and developers looking to enhance their teaching methods.
- ITS reduce the dependency on human resources
- ITS help students more in understand material by permitting them to first state what they know, then by outfitting responses accordingly
- ITS afford educators the chance to produce individualized programs due to their tailored nature.
- ITS harvest higher exam marks than traditional systems, particularly in students from special education, non-native English, and low-income backgrounds.
- ITS Provide immediate customized feedback, individual task selection, on-demand hints, and provision for mastery learning.

Some of the criticism of Intelligent Tutoring Systems includes:

- ITS is difficult to measure its effectiveness
- ITS immediate feedback and hint sequences fail to improve deep learning in students.
- ITS fail to ask questions of students that may clarify their actions.
- ITS implementation may be hard to defend to an administrative staff.
- ITS evaluation often hard, costly and time consuming.
- Human tutors are currently better than ITS in providing proper dialogue and feedback.
- Human tutors are currently better than ITS in interpreting and adapting to different emotional states.

LITERATURE REVIEW

An Intelligent tutoring system provides customized feedback instruction to students [23,33]. There are many ITS system designed for educational purposes [8-15,25], an agent based ITS for Parameter Passing in Java Programming[17], Java Expression Evaluation [13], Linear Programming[10,31], an Intelligent Tutoring System for Entity Relationship Modeling[16], an Knowledge-based Intelligent Tutoring System for Teaching Mongo Database[22], ITS for learning Software Patterns [7], Design and Development of an Intelligent Tutoring System for C# Language[32], ITS called JO-Tutor for helping students to learn Java Programming language [28], ITS which called CPP-Tutor for helping student to learn C++ Programming Language[14], a comparative study between Animated Intelligent Tutoring Systems (AITS) and Video-based Intelligent Tutoring Systems (VITS) [24], effectiveness of e-learning[26], computer aided instruction[9], effectiveness of the CPP-Tutor[29], teaching AI searching algorithms[15], teaching database to sophomore students in Gaza[12], and Predicting learners performance using NT and ITS [8], design and development of diabetes ITS[30], ITS teaching grammar English tenses [27], ITS for teaching advanced topics in information security[19], development and evaluation of the Oracle Intelligent Tutoring System (OITS)[20], ITS for learning Computer Theory[21], e-learning system[11,18,27].

ITS ARCHITECTURE

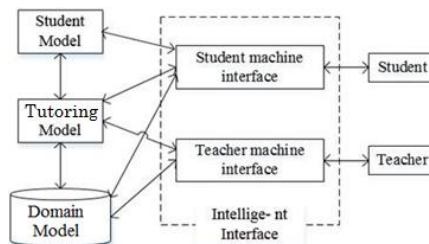


Figure 1: Architecture of ADO-Tutor

Modern intelligent tutoring system architectures consist of four modules as in Figure 1. The four modules are represented normally as the domain module, student module, tutoring module and the interface module[7,12].

The domain module is how the material is represented and stored to be ready for transmitting to the learner.

The domain module is connected to the student module and the tutoring module. The student module represents the learner's conduct such as his learning style, motivation level, his interests, and profile. All the student conduct is logged into the system and used for reasoning and adapting the domain module to the learner's needs [15].

The tutoring module acts as a computer-generated instructor, presenting the material in an proper order, according to the student learning style and his skills. This is a collaborating process and this module has the task to clarify the concepts to the learner given few points of view and supportive all the learning procedure [19].

With the capacity to interconnect and cooperate with the student, the interface module has an enormously significant job. If one ITS have great tutoring, student and domain modules, but the interface module is very deprived, the ITS will not be operative because the interface is the front of whole system and has the aptitude to capture all the care of the learner [21].

DOMAIN MODEL ARCHITECTURE

The domain model of ADO-Tutor includes the following material:

- Lesson 01: Introduction to ADO.NET
- Lesson 02: The SqlConnection Object
- Lesson 03: The SqlCommand Object
- Lesson 04: Reading Data with the SqlDataReader

- Lesson 05: Working with Disconnected Data – The DataSet and SqlDataAdapter
- Lesson 06: Adding Parameters to Commands
- Lesson 07: Using Stored Procedures

STUDENT MODEL ARCHITECTURE

The system store the student data in a database, and by recording his own name, number, last session time and date, lesson learned, current score, overall core and all the details of the student or the learner of the system.

TEACHER MODULE ARCHITECTURE

The storage information about the user can used to customize the feedback to display the relevant information to students about the educational level and stage passed the student and how much he/she studied, and difficulty level reached. For example, when the student's current score out of 100 reaches 75 or more, the tutoring module takes the student to next level of difficulty and give the student a notification about the score he/she achieved.

USER INTERFACE MODEL

The tool used for building the ADO-Tutor has two interfaces in one. The first is the teacher interface, where he/she can add the lessons, questions, answers, level of difficulties, basic ITS data, Basic student data, and customize the coloring of the screen of the system. The second interface is the student interface where he/she is allowed to the leering material screen, questions screen, statistics about his performance screen as seen Fig 2-Fig5.



Figure 2: In this interface student chooses the desired lesson to learn

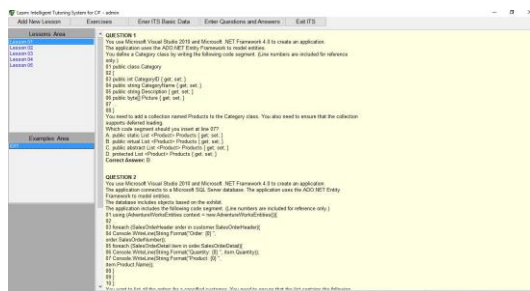


Figure 3: In this interface student chooses examples required to take advantage

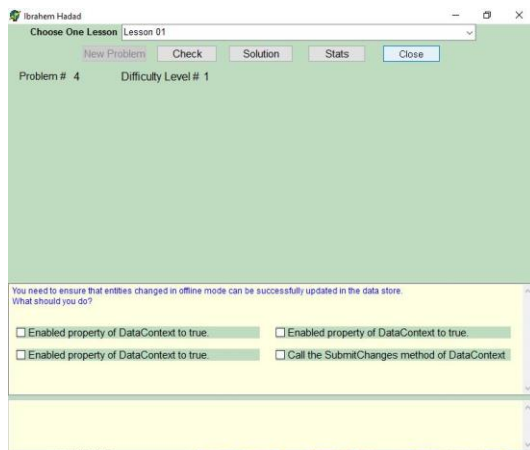


Figure 4: In this interface student chooses question to try and answer it.

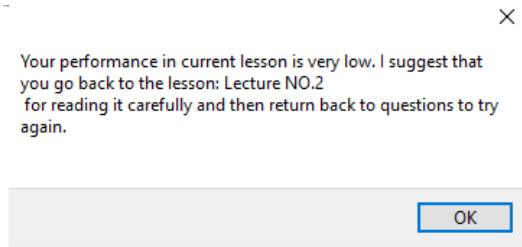


Figure 5: This is a special notification about the progress of the student

EVALUATION

DO-Tutor was presented to a group of specialists in the field to evaluate it and gives us feedback about the design of the system, the material offered to the learners, easiness of use, quality and expressiveness of the exercises and their relation to the lessons given to the learners. The results of the evaluation were promising. We will do some more evaluation in the near future to check the effectiveness of the ADO-Tutor and take into considerations the comments of the evaluators.

CONCLUSION

In this paper, we created an intelligent tutoring system called ADO-Tutor for teaching database connection. ADO-Tutor was designed and built using ITSB authoring tool. Through what has been reviewed we have created a simple and easy way to explain the material instruction, while ensuring that the student take advantage of through questions and find solutions for those who could not answer. An initial evaluation was done by a set of specialist in the field and the results were promising. In the future, we will increase the lessons to cover a greater range of educational material.

REFERENCE

1. www.dotnettricks.com
2. ADO.NET Code Examples - MSDN - Microsoft
3. Database Programming - ADO.NET
4. Database Programming using ADO.NET and C#
5. [Introduction to ADO.NET - SitePoint](#)
6. Create a Database with ADO.NET - Xamarin
7. Jeremić, Z. (2005). An Intelligent Tutoring System for learning Software Patterns (in Serbian), Master Thesis, Belgrade, Serbia.
8. Naser, S. S. A. (2012). Predicting learners performance using artificial neural networks in linear programming intelligent tutoring system, International Journal of Artificial Intelligence & Applications, Academy & Industry Research Collaboration Center (AIRCC), 3,2, 65-73.
9. Naser, S.S A., & Sulisel, O. (2000). The effect of using computer aided instruction on performance of 10th grade biology in Gaza.
10. 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.
11. Chrstan Wolf, "I-Weaver: Towards learning style-base e-learning in computer science education", proceedings of the Australian Computing Education Conference, 2003.
12. 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.
13. Abu Naser, S. (2008). JEE-Tutor: An Intelligent Tutoring System for Java Expression Evaluation, Information Technology Journal, Scialert , 7(3),528-532.

14. 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.
15. Naser, S.S.A. (2008). Developing visualization tool for teaching AI searching algorithms, *Information Technology Journal, Scialert*,7(2), 350-355.
16. Pramuditha Suraweera & Antonija Mitrovic. KERMIT, An Intelligent Tutoring System for Entity Relationship Modeling. Private Bag 4800, Christchurch, New Zealand
17. Naser, S. A. (2008). An Agent Based Intelligent Tutoring System For Parameter Passing In Java Programming, *Journal of Theoretical & Applied Information Technology*, 4,7.
18. Mahmoud, A. Y., Barakat, M. S., & Ajjour, M. J. (2016). Design and Development Of ELearning University System. (*Journal of Multidisciplinary Engineering Science Studies (JMESS)*), 2(Issue 5, May - 2016), 498-504.
19. 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.
20. ALDahdooh, R., & Abu Naser, S.S. (2017). Development and Evaluation of the Oracle Intelligent Tutoring System (OITS), *European Academic Research*, 4(10).
21. Al-Nakhal M.A., & Abu Naser, S. S. (2017). An Intelligent Tutoring System for learning Computer Theory, *European Academic Research*, 4(10).
22. Hilles, M. M., & Abu Naser, S.S. (2017). Knowledge-based Intelligent Tutoring System for Teaching Mongo Database, *European Academic Research*, 4(10).
23. Abu Naser, S. S. (2016). ITSB: An Intelligent Tutoring System Authoring Tool. *Journal of Scientific and Engineering Research*, 3(5), 63-71.
24. 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.
25. 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.
 26. 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 (IJDKP),1, 1-13.
 27. 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).
 28. 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 (IJAlA), 2(2).
 29. 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/>.
 30. Almurshidi, S. H., & Abu Naser, S. S. (2016). Design and Development of Diabetes Intelligent Tutoring System, European Academic Research, 4(9).
 31. 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)
 32. Al-Bastami, B.H., & Abu Naser, S.S. (2017). Design and Development of an Intelligent Tutoring System for C# Language, European Academic Research, 4(10).

33. 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).
34. Abu Naser, S. (1993). A methodology for expert systems testing and debugging, North Dakota State University, USA 1, 1-130.
35. Naser, S.S.A. (1999). Big O Notation for Measuring Expert Systems complexity, Islamic University Journal - Gaza 7 (1), 77-57.