Available online www.jsaer.com

***Journal of Scientific and Engineering Research*, 2016, 3(5):63-71**

ISSN: 2394-2630
CODEN(USA): JSERBR

Research Article

ITSB: An Intelligent Tutoring System Authoring Tool

Samy S Abu Naser

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

Abstract Intelligent Tutoring System Builder **(**ITSB) is an authoring tool designed and developed to aid teachers in constructing intelligent tutoring systems in multidisciplinary fields. The teacher is needed to create a set of pedagogical fundamentals, which, in line, are inured to automatically build up a broad tutor framework and construct an intelligent tutoring system. In this paper an explanation of the theory and the architecture of the tool are outlined. A presentation of several system components, the requirements of the different components, integration of these components in ITSB tool are shown. Furthermore, implanting of requirements, cognitive principle, and common design fundamentals in the tool to ease the use of teachers. A variety of design matters, an example of building an intelligent tutoring system for teaching Java language using ITSB tool and an evaluation are presented.

Keywords ITSB, Intelligent Tutoring Systems, Java Language

Introduction

Intelligent Tutoring System (ITS) is a software that aims to provide immediate and customized instruction or feedback to learners [1-14], typically without interference from a human teacher. ITSs have the general aim to facilitate learning in a evocative and efficient way by using a diversity of computing technologies. There are a lot of examples of ITSs being used in both official education and professional situations in which they have confirmed their abilities and boundaries. There is a strong association between intelligent tutoring, cognitive learning theories and design. ITS intends to solve the problem of over-dependency of students over teachers for superiority education. It intends to offer access to high-class education to every student, consequently improving the whole educational system.

The aim of ITS authoring tool is to simplify the development of construction ITSs and reduce the ability threshold for constructing them. In addition, they permit the quick prototype of ITS design. Accomplishing these objectives will assist ITS developers and users who do not have computational thoughts construct ITSs. Computational thoughts concerned with creating problems and their solutions so that the solutions are formulated can efficiently carried out by an information processing agent.

Literature Review

The Cognitive Tutor Authoring Tool allows a teacher to include learning by doing to online lessons [15]. It facilitates the creation of two categories of tutors: example-tracing tutors that can be produced with no programming however it needs problem-specific authoring, and cognitive tutors, that need Artificial Intelligence programming to create a cognitive model of student problem solving however, provide tutoring across a variety of problems [15].

The Extensible Problem Specific Tutor [16] is web authoring tool that assists quickly developing example-tracing tutors on existing interfaces that reduces tutor development time, and permits the interface to be

separable from the tutoring component. The Presentation Manager gives visual feedback to the student on the interface.

RIDES is an authoring tool for creating and delivering graphical simulations and simulation-based training. With RIDES, the author can convert a set of simulations into interactive, graphical tutorials. Actually, the reason why RIDES might never be used for reproducing Ms. Lindquist is due to its limitation to construct tutors based on simulations only [17].

ASPIRE an authoring tool for assisting with the creation and delivery of constraint-based tutoring systems. It produces constraints that structure the domain model with the help of the domain expert, reducing the programming expertise necessary for developing a new constraint-based tutor. The system also offers all the domain-independent functionality of constraint-based ITSs [18].

ITSB Authoring Tool Overview

ITSB authoring tool is a shell for crating intelligent tutoring systems. It is designed and developed using Delphi Embarcadero XE8, 2015; ITSB authoring tool is two systems in one application. The first one is the teacher system where he/she add the course materials, questions and answers etc. and the second system is the students where he/she learn the course material and practice exercises. The following sections describe the system architecture and next an illustration of the different modules, their purpose, functions, and goals etc are outlined.

ITSB System Architecture

A normal ITS has four fundamental modules: domain model, teaching model, student model and user interfaces. The domain model adds the course configuration in a structured style. A course may have a variety of parts, such as division, sub-divisions, and topics. These parts are stored in the domain model together with their dependencies. All the materials and resources necessary to tutor a student are also kept in this module.

The student model is the demonstration of the students the system is coping with. The student model provides the system with all required information so it can adapt itself with the student. Therefore, student model is a vital tool for the adaptation process.

The teaching module contains all the decision-making procedure concerning course preparation and adaptation. Often, this module is called the control engine, because this module controls the entire system, by accepting inputs from the other parts.

Lastly, the user interfaces have two sections - one for the student and the other for the teacher. Teacher's interface is accustomed to arrange and adjust the system and its different parts. So, the teacher's interface behaves as the authoring tool. By his interface, the teacher can add new lessons, adjust the established ones, and revise teaching methods. The student's interface is used to convey all the teaching commands. The sort and the type of these commands would differ with student's ability and performance level. A general system architecture is shown in Fig. 1.

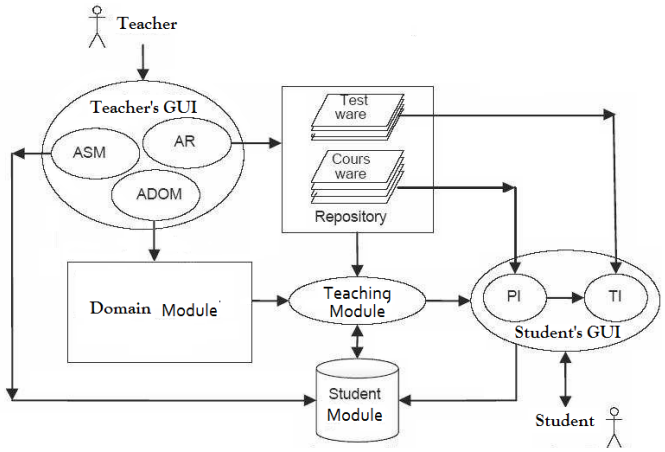

Figure 1: Overall System Architecture of ITSB

ITSB Domain Model

The domain model is concerned with the lessons, its arrangement and a range of elements. There are two fundamental components in domain model.

The first component, Domain Organization Model, deals with the arrangement and organization of the lessons and its topics. The second one, Repository**,** deals with the materials being taught themselves.

ITSB Student Model

State based approach was implemented in the student model. However, there are quite a few parameters for educational modeling of a student throughout a learning procedure. Two parameters were taken into account in this paper:

- Coverage: the topics covered by a student and
- Performance of a student: (measured through his ability to comprehend and his problem solving skills.

ITSB Teaching Module

Teaching module is considered to be the most important component of an ITS. The primary task of this module is to arrange a sequence of teaching actions to be taken during a teaching process. These actions and their sequence should go with the student's ability, requirement and objectives.

The arrangement is done at two stages. At the first stage, ordering of the topics for the student needs to be arranged. This stage begins from the initial state and finishes when all the topics are included in the sequence. At the second stage, after a topic is chosen another arrangement is essential to compute the exact technique of teaching that topic. This engages selecting the proper type of the document and the proper medium.

ITSB User Interfaces

Interfaces are an essential part of the ITSB system. There are two classes of users, teachers and the students. The ITSB authoring tool has both interfaces. Each class of users see different interface for their interactions with the system. The teachers interface is the shell of ITSB for configuration and adjustment of the system. The teacher's interface or the authoring interface consists of three parts, used to configure the different parts of the system, one to configure the Student Model, one for authoring the Domain Organization Model and the third for maintaining the Repository (see Fig. 1). Through these interfaces a teacher can configure various aspects of the system, like initial information about the student , enter students lessons, questions and answers, configure and adjust the color, font name and size of all menus, buttons, comb boxes etc. Thus, this interface provides the system with the required flexibility and robustness. Moreover, due to this interface the system can become domain independent. A screenshot of the teacher's interface is shown in Fig 2, Fig 3, Fig 4 and Fig 5.

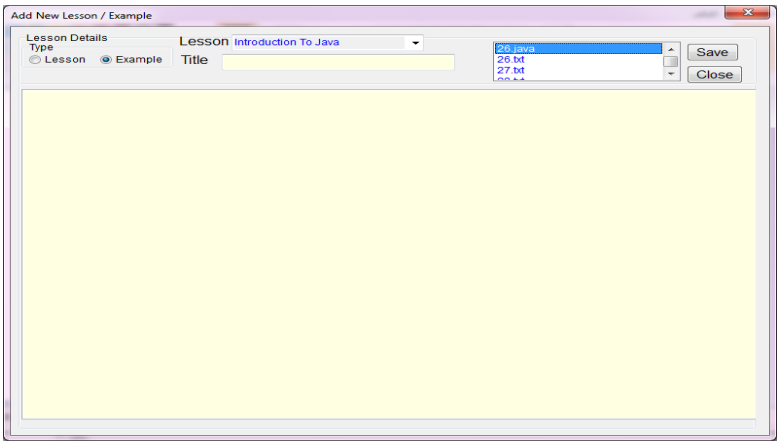

Figure 2: Form for adding Lessons and Examples

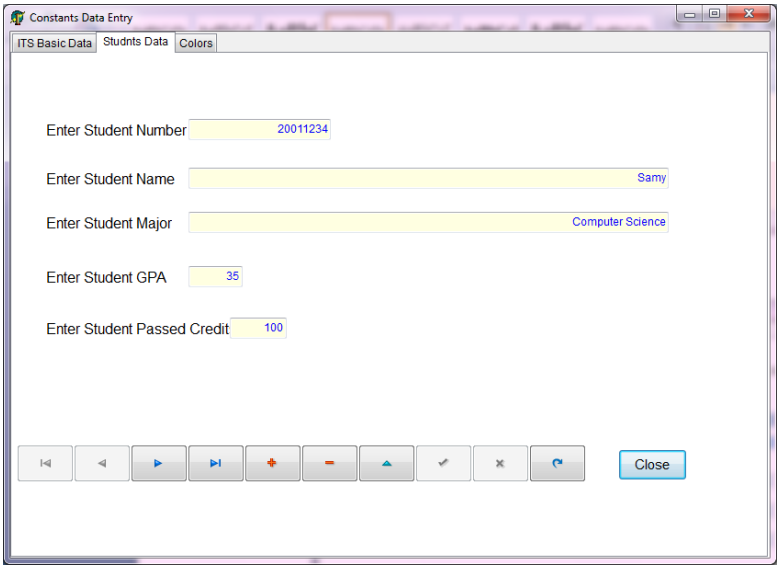

Figure 3: Form for adding initial students information

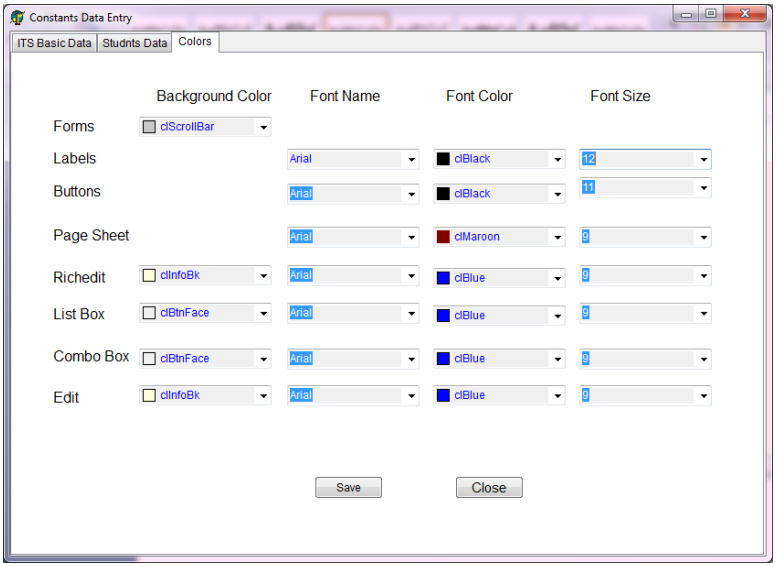

Figure 4: Form for adjusting Fonts of all screens of the system

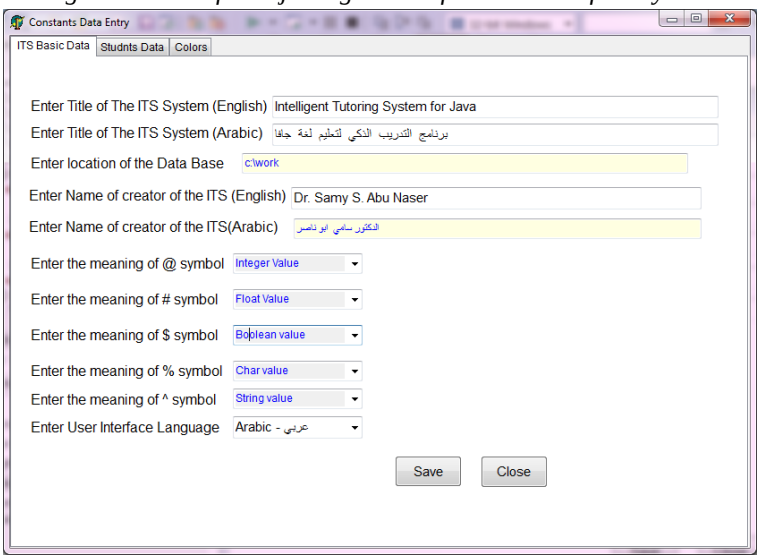

Figure 5: Form for adding constants of the system

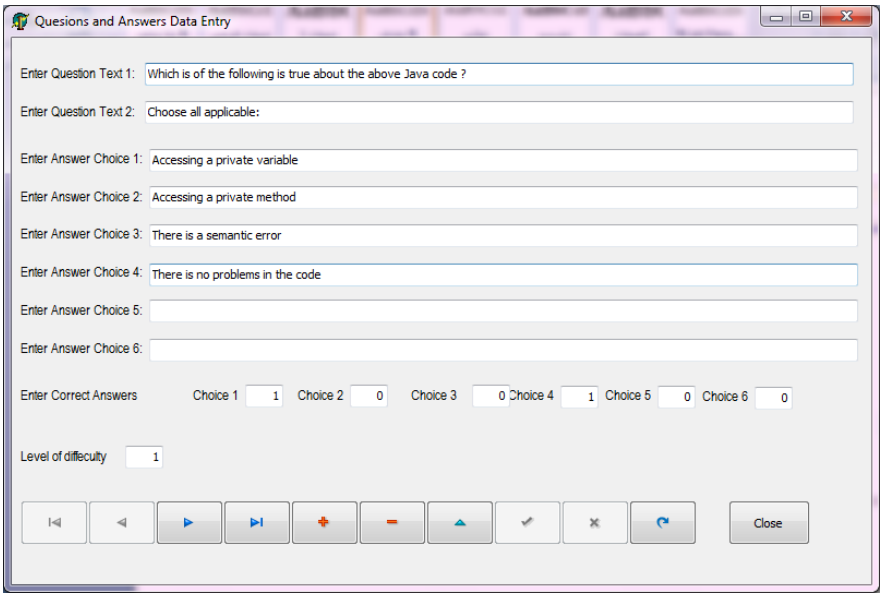

Figure 6: Form for adding questions and answers

Student interface is the front-end for the student to interact with the system. The interface has a bidirectional communication mechanism (see Fig. 1). The system presents all the learning documents and test materials to the student through this interface. Performance of the student in the tests is conveyed back to the system, specifically to the student model by it. This feedback is vital because the adaptation process would depend on this. So, the success of adaptive planning depends on it and its communication with teaching module (See Fig 7, Fig 8, Fig 9).

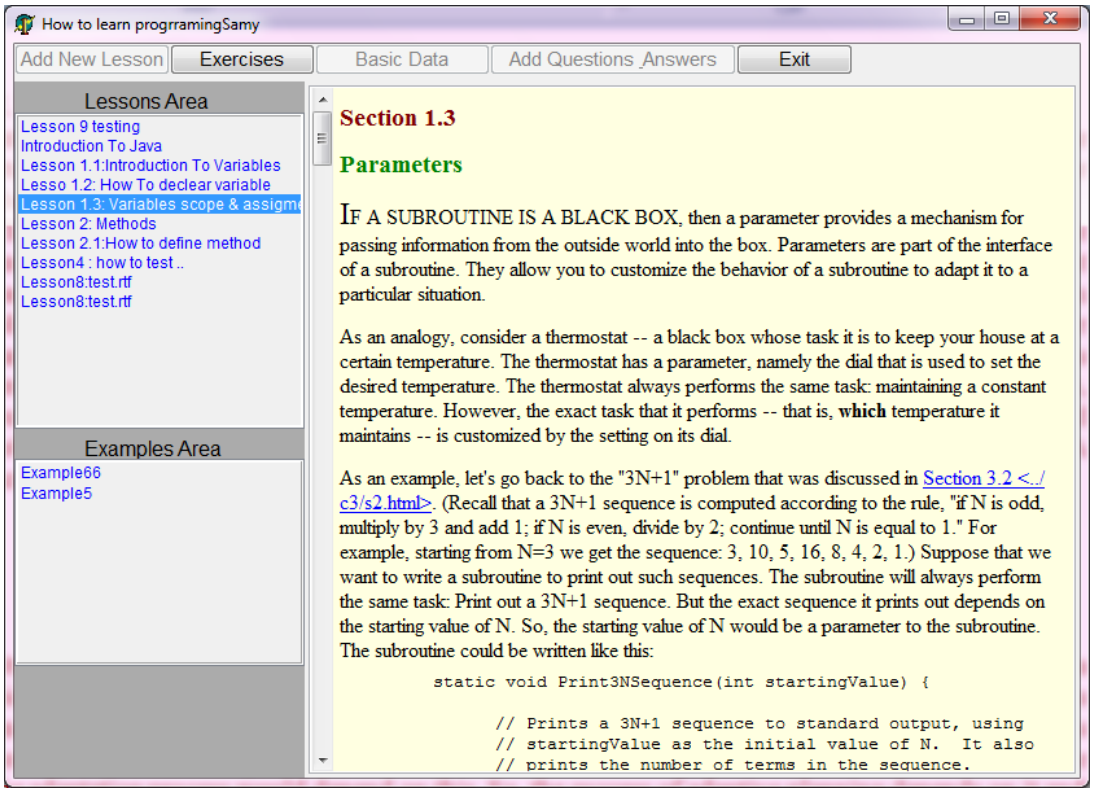

Figure 7: Student lessons and examples form

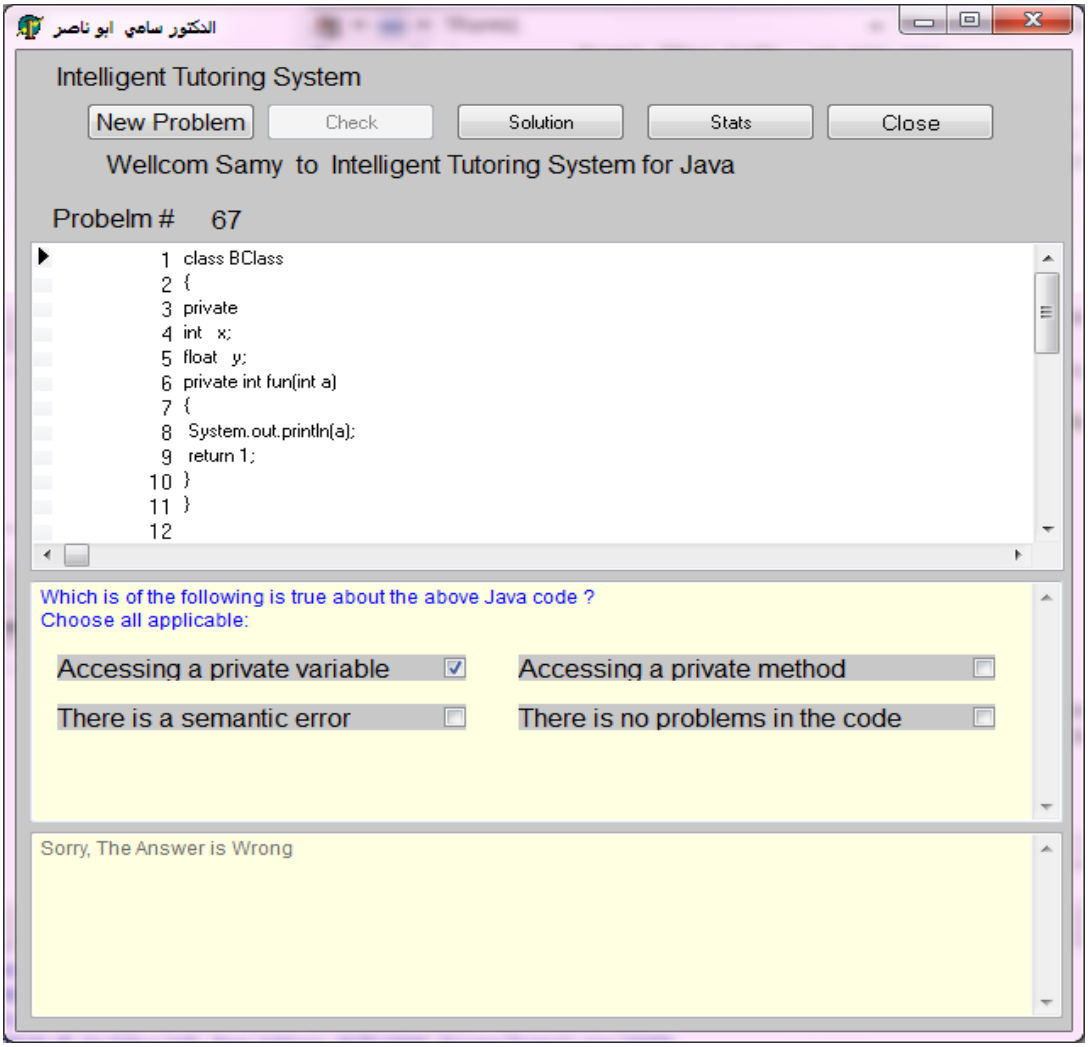

Figure 8: Student Exercises form

Apart from these, the student interface also provides various courses and performance related information to the student, such as the performance history of the student, how much of the course has been covered, detailed test results etc. These help a student to identify and rectify his status (See Fig 9).

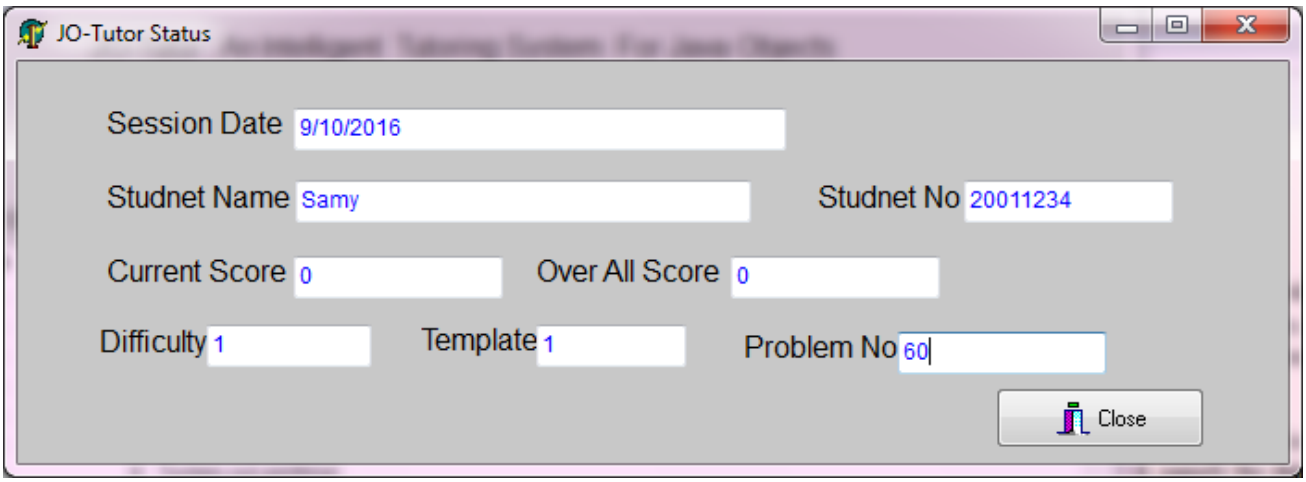

Figure 9: Student statistics form

Multiple Language Support

ITSB authoring tool currently support two languages. The default language is English language. If the user prefer Arabic Language, He/she can just click on one button in first Login form. Once the user clicks the Arabic button, it translate all buttons, menus, titles, and subtitles; furthermore it switch the direction of the forms from Left-to-Right into Right-to-Left (See Fig 10, Fig 11).

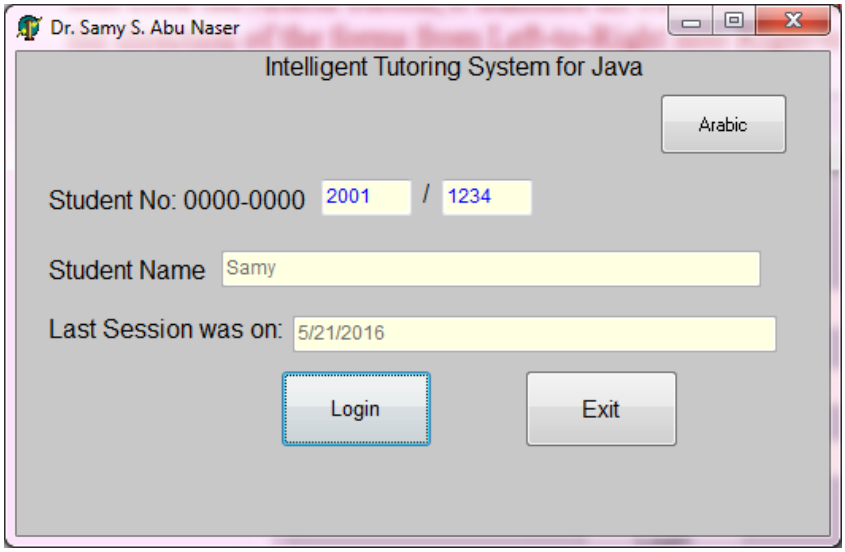

Figure 10: Logging Form in English Language

Figure 11: Logging Form in Arabic language

ITSB Evaluation

Two evaluations were carried out to test the ITSB intelligent tutoring system authoring tool.

The first evaluation was to let a group of teachers each prepare the materials: lessons, examples, exercises, answers, student information, and system constants etc. for introductory java programming. Then each teacher was asked to enter the prepared materials using a copy of the ITSB tool. The next step was to gather the opinion of each teacher in team of how easy, efficient, and friendly was the ITSB tool.

The second evaluation was to let a group of students use one of the resultant intelligent tutoring system to see how easy, efficient, and friendly.

The result of the first evaluation (teachers evaluation) and second evaluation (students evaluation) are promising as shown in Table 1.

Table 1: The results of both evaluations

Category	Teachers Evaluation Results	Students Evaluation Results
The system is easy to use	88%	92%
The System is efficient	85%	90%
The System is friendly	82%	93%

Conclusion

In this paper we have described the theory and architecture of the ITSB tool. ITSB tool is an authoring tool designed and developed to aid teachers in building intelligent tutoring systems in multidisciplinary fields. Several system components, requirements of the different components, and integration of these components in ITSB tool were outlined. Furthermore, implantation of requirements, cognitive principle, and common design fundamentals in the tool to ease the use of teachers. A variety of design matters, and an example of building an intelligent tutoring system for teaching Java language using ITSB tool was outlined. Two evaluations of the teachers and students perspective were presented and the results were promising.

References

[1]. Naser, S.S.A., (2008). Developing an intelligent tutoring system for students learning to program in C++. Information technology journal, 7(7), pp.1055-1060.

[2]. Naser, S.S.A., (2008). Developing visualization tool for teaching AI searching algorithms. Information Technology Journal, 7(2), pp.350-355.

[3]. 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), p.65.

[4]. 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, 5(5), pp.916-922.

[5]. 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), pp.109-114.

[6]. Naser, S.A., (2008). An agent based intelligent tutoring system for parameter passing in java programming. Journal of Theoretical & Applied Information Technology, 4(7).

[7]. Abu-Naser, S., Ahmed, A., Al-Masri, N., Deeb, A., Moshtaha, E. and AbuLamdy, M., (2011). An Intelligent Tutoring System for Learning Java Objects. International Journal of Artificial Intelligence and Applications (IJAIA), 2(2).

[8]. Abu-Naser, S., Al-Masri, A., Sultan, Y.A. and Zaqout, I., (2011). A prototype decision support system for optimizing the effectiveness of elearning in educational institutions. International Journal of Data Mining & Knowledge Management Process, 1(4).

[9]. Naser, S.S.A., (2008). JEE-Tutor: An Intelligent Tutoring System For Java Expressions Evaluation. Information Technology Journal, 7(3), pp.528-532.

[10]. 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), p.209.

[11]. Naser, S.A., Ahmed, A., Al-Masri, N. and Sultan, Y.A., (2011). Human Computer Interaction Design of the LP-ITS: Linear Programming Intelligent Tutoring Systems. International Journal of Artificial Intelligence & Applications (IJAIA), 2(3,), pp. 60-70.

[12]. Abu Naser, S.S., (2001). A comparative study between animated intelligent tutoring systems AITS and video-based intelligent tutoring systems VITS. Al-Aqsa Univ. J, 5, pp.72-96.

[13]. Abu Naser, S.S. and Sulisel, O., (2000). The effect of using computer aided instruction on performance of 10th grade biology in Gaza. J. Coll. Edu, 4, pp.9-37.

[14]. Abu Naser, S., (2008). JEE-Tutor: An Intelligent Tutoring System for Java Expression Evaluation. Information Technology, 7(3), pp.528-532.
[15]. Koedinger, K. R., Aleven, V. A., & Heffernan, N. (2003). Toward a rapid development environment for cognitive tutors. *Artificial intelligence in education: Shaping the Future of Learning Through Intelligent Technologies*, *97*, 455.
[16]. Gilbert, S., Blessing, S. B., & Kodavali, S. (2009). The Extensible Problem-Specific Tutor (xPST): Evaluation of an API for tutoring on existing interfaces. Paper presented at the Artificial Intelligence in Education.
[17]. Allen Munro, Mark C. Johnson, Quentin A. Pizzini, David S. Surmon, Douglas M. Towne, James L. Wogulis (1997). Authoring Simulation-Centered Tutors with RIDES. International Journal of Artificial Intelligence in Education, 8, 284-316.
[18]. Mitrovic, A., Suraweera, P., Martin, B., Zakharov, K., Milik, N., & Holland, J. (2006). Authoring constraint-based tutors in ASPIRE. Paper presented at the 8th International Conference on Intelligent Tutoring Systems.