

A Suggested Generic Intelligent Tutoring Framework

Fares Fraij* and Victor Winter**

**Department of Information Technology
Al-Hussein Bin Talal University
P.O.Box 20
Ma'an, Jordan
fares@ahu.edu.jo*

***Department of Computer Science
University of Nebraska at Omaha
6001 Dodge Street, PKI 175A
Omaha, NE 68182-0116
USA
vwinter@mail.unomaha.edu*

Abstract: This paper presents an *intelligent tutoring framework* that can be effectively utilized to assist teaching courses and therefore to achieve pedagogical goals. The courses generated using the framework are *adaptive*, i.e., they adjust their behavior to overcome the individual differences among students. The architecture of the framework provides three modules for an administrator, an instructor and a student. Furthermore, students explore the material of the course through two modes, namely *non-interactive* and *interactive (or adaptive)*. To achieve the goals of the framework, it is recommended to employ an agile software development process such as extreme programming. Furthermore, the development team of the framework must involve students and therefore proceeds in a *user-centered* fashion.

Key words: Extreme Programming, Interaction, Intelligent Tutoring Systems, User-Centered Approach.

1. Introduction

An Intelligent Tutoring System (ITS) is a software application for delivering course material. Its purpose is to improve learning outcomes. To be effective, such systems must be adaptable to individual differences among students. Adaptability can be achieved by empowering an ITS with the ability to adjust its behavior in order to provide the students with a flexible and efficient learning environment.

ITS can be either *static* or *dynamic*. A static ITS is course specific, i.e., it is designed and implemented to teach a specific course. As a result, the benefits of a static ITS is somewhat limited. In contrast, a dynamic ITS is course independent, i.e., such a system is designed to teach any course. Such ITS are advantageous as they can be loaded with the material of any course and automatically generate a learning environment. In this paper, we will use the term *dynamic ITS* and *intelligent tutoring framework (ITF)* interchangeably.

In general, an ITS can be employed in blended learning [1] or e-learning [2]. Blended learning is characterized by introducing course material with a mix of traditional face-to-face approach and technology [1]. Employing technology may take several forms such as presenting visual activities to

elaborate on some concepts of the material using a PowerPoint presentation, a computer, and a data show, or using a software application. The goal of e-Learning is to provide education at a distance. It is worth mentioning that fully-taught courses through e-Learning may not be accredited in some countries.

Although commercial computer supported instruction environments are available such as WebCT [3] and Blackboard [4, 5], these environments are costly to afford. Furthermore, such environments are equipped with many functions that may not be fully utilized by the instructors.

Adaptive computer-aided Systems have been investigated in the literature [7, 8, 9]. In [7] an adaptive e-Learning framework is introduced, which dynamically generates suitable courses for each student. In [8], a framework for adaptive e-Learning based on distributed reusable learning activities was investigated by utilizing a Knowledge Tree. In [9] a preliminary assessment of the adequacy of existing e-Learning standards for supporting the introduction of adaptation techniques in e-Learning systems was discussed. Interesting shortcomings identified in the available e-Learning standards were insufficient coverage of the model of the individual learners and the need to enrich the methods of adaptation of e-Learning systems.

This paper presents an intelligent tutoring framework intended to be employed in either blended learning or e-learning. The paper is organized as follows. Section 2 introduces the pedagogical underpinnings of the framework. In section 3, the architecture of the framework is discussed. Section 4 introduces the modes of interaction supported by the framework. Section 5 presents the software development process to be employed in constructing the framework. Section 6 concludes and discusses future directions.

2. Pedagogical Underpinning

Introductory courses offered at a university may attract students with a relatively broad range of background. This background diversity presents challenges to instructors. On one hand, effort must be made to provide students with a common foundation upon which the more advanced topics of the course will be built. On the other hand, if too much time is spent on foundational (prerequisite) material, the quality of the course will suffer and the more advanced students will not be sufficiently challenged.

A long standing rule-of-thumb in teaching is to address individual differences among students in order to maximize learning opportunities. To achieve this goal, educators try to adapt learning material and curriculum to meet individual differences among the students [10]. Adapting the material to a particular student's background requires measuring the student's knowledge of the material. This can be achieved by interacting with the student which provides a student-centered collaborative environment. Historically, Socratic Method employed questions as a mean to interact with students to achieve educational goals [11].

An *evaluation test* (or pre-test) [10] represents a convenient mechanism for measuring a student's grasp of a particular subject matter. Within the framework of a course, such pre-tests are useful for evaluating a student's knowledge prior to presenting specific material. The benefits of such tests are that they represent foundation of instructional decisions, help an instructor how best to present a given subject matter and influence the nature of interaction between students and the instructor. Pre-tests also represent efficient, yet unbiased means to know what the students are capable of knowing or doing. The instructor can plan more effective material and the students are helped to learn more [12].

It is also crucial for instructors to ensure that students have achieved the learning objectives of a course. In this regard, post-tests or summative evaluation [10] are used to measure what the students have learned after the instruction is completed. Recently, higher levels of interaction can be achieved due to the employment of technology in teaching. It is important to realize that recent advances in ICT has relieved educators concerns such as keeping records of

pre- and post-tests [10] and preparing and conducting comprehensive pre- and post-tests [12]. It is worth mentioning that incorporating visual activities during presenting the course material help in increasing students' interaction in a teaching session.

3. Framework Architecture

Figure 1 presents the architecture of a framework which consists of three modules: an administrator, an instructor and a student. The administrator is responsible for creating usernames and passwords for instructors and is in-charge of the evolvement of the framework. Instructors feed-in the course material and plan the lessons of the course. Finally, students access and interact with the material of the course and submit comments and questions to their instructors. All the users of the system are connected to a central database.

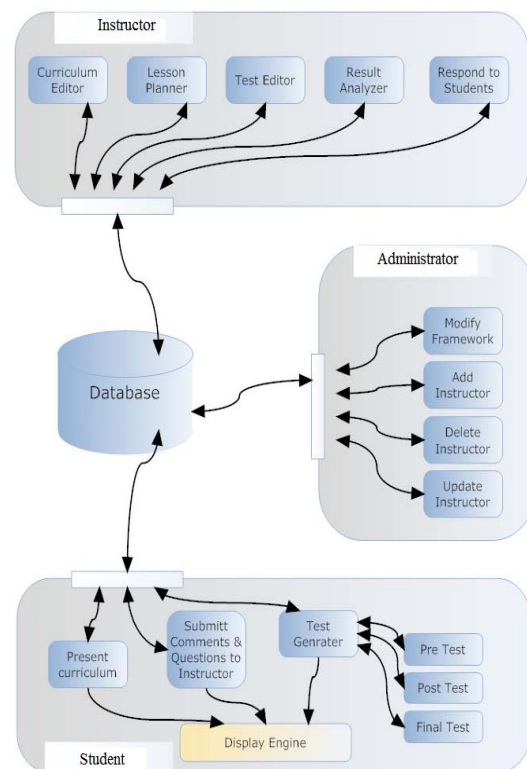


Figure 1. Overview of the Framework Architecture

3.1 Administrator Module

The administrator of the framework receives feedback from users, studies them, and performs the necessary modifications to the framework. The administrator is also responsible for detecting and fixing programming bugs. Finally, the administrator creates accounts for instructors interested in using the framework to deliver their courses.

3.2 Instructor Module

The instructor performs major activities such as preparing the material of a course by dividing it into lessons where each lesson consists of sections [13].

Each section is atomic in the sense that it only represents one concept and is associated with well-defined set of skills a student is required to achieve by the end of the section. The *lesson planner* is a tool that is used by the instructor to determine the structure and the sequence of the lessons of the course. The *test editor* module is used by the instructor to construct a database of questions to be used in tests. The *result analyzer* tool is mainly built to obtain statistics from student's performance database. The tool considers the students' profile to study their performance and to identify difficulties faced by students belonging to a variety of majors.

3.3 Student Module

Students interact with the framework through the *display engine*. The students have a username and a password to login in to the engine. Through the engine, the students view and interact with the material of the course and perform the needed exams.

4. Modes of Interaction with the User

The courses developed by the framework can be presented in two modes: *non-interactive* and *interactive*. The algorithm of the two modes was thoroughly discussed in [14]. In addition, a course specific implementation of a simplified version of the framework was presented [15]. However, the suggested framework introduced in this paper is course independent and designed to present any course of choice. For completeness, the two modes are briefly discussed in this section.

4.1 Non-Interactive Mode

This mode presents the course's material in a predetermined, recommended order similar to the traditional teaching method. The material's concepts are presented in a menu so that a student can choose the concepts s/he wants to consider. By the end of each concept, a set of exercises is introduced, and finally a *post-test* is proposed. The grade of the post-test is recorded and the process is repeated until all concepts are covered. If the student performance is good in all the post-tests of the concepts, the student is introduced to a final post-test to measure his/her performance. Otherwise, a menu is formulated and presented to the student containing the concepts s/he faces challenges with again.

4.2 Interactive Mode

This mode encourages a student to focus on the concepts s/he does not know and the student is given the opportunity to skip to the concepts s/he already familiar with. To achieve this goal, the student's grasp of the course concepts is measured before introducing him/her to the course concepts through a *pre-test*. Afterward, a menu is formulated and presented to the student. This menu focuses on strengthening the

concepts the student found challenging.

5. Development Process

In spite of the substantial development costs and development time associated with producing quality eLearning applications [16], a recent study [6] has shown that the development cost of such applications can be dramatically decreased by the availability of well-trained IT staff co-operate to develop in-house software and employ a rapid software development process such as extreme programming [15].

The proposed framework will be developed using Microsoft.NET Framework [17]. More specifically, VisualBasic.NET [18] will be employed as the main software development tool. The reasons for choosing VB.NET is the availability of VB.NET programmers and its adequacy and support for implementing web-based applications.

A software development team will be developed using a *user-centered* approach [19]. This approach directly studies the cognitive, behavioral, and attitudinal characteristics of users. Furthermore, the users' reactions and performance to scenarios, manuals, simulations & prototypes are observed, recorded and analysed. The user will be a full-time, long-term member of the project team which will be introduced in the next paragraph.

The team will also have a team leader, an analyst, a designer, a programming specialist, a design specialist, and educational specialist. The framework will be developed using an agile software development process, namely extreme programming. This method is appropriate for interactive software design, has been proven to be successful expediting the time-to-market [20], and reduces risk associated with the conformance between implementation and user requirements.

6. Conclusion and Future Directions

An adaptive computer-supported framework for generating courses has been sketched in this paper. The framework is oriented toward adjusting its behavior to fit student's needs. The framework also is intended to enable instructors with limited IT skills to develop e-courses. The framework is course independent and is recommended to be employed in teaching courses within a university. When used in conjunction with traditional face-to-face teaching, the framework is expected to improve the learning outcomes as it handles individual differences among students in the classroom.

References

- [1] The Ohio State University (OSU) "E-Learning Implementation Strategy and Plan," 2003. <http://telr.osu.edu/resources/ITelearning.pdf> accessed January, 12 2008.

- [2] M. Ramshirish and P. Singh, "E-Learning: Tools and Technology," DRTC Conference on ICT for Digital Learning Environment, Bangalore, India, January, 2006.
- [3] M. W. Goldberg and S. Salari, "An Update on WebCT (World-Wide-Web Course Tools) A Tool for the Creation of Sophisticated Web-Based Learning Environments," Proceedings of NAUWeb, June 12-15, 1997, Flagstaff, Arizona.
- [4] Blackboard Inc., Blackboard Learning System, 2004.
- [5] Blackboard Inc., Educational Benefits of Online Learning, 2000, <http://www.blackboard.com>.
- [6] A. AlDmour and F. Fraij, "Developing In-House Software: Seeking for Excellence, Facing Challenging, and Exploiting Opportunities", ICITE, Paris, France, 2008.
- [7] M. Sasakura and S. Yamasaki, "A Framework for Adaptive e-Learning Systems in Higher Education with Information Visualization," 11th International Conference Information Visualization (IV'07), 2007.
- [8] P. Brusilovsky and H. Nijhavan "A framework for adaptive e-learning based on distributed re-usable learning activities", Proceedings of World Conference on E-learning, 2002, pp.154-161, 2002.
- [9] A. Paramythis and S. Loidl-Reisinger "Adaptive learning environments and e-learning standards", Electronic Journal of eLearning, 2(1), March, 2004.
- [10] R. M. Thorndike, Measurement and Evaluation in Psychology and Education, 7th Edition, Prentice Hall, 2004.
- [11] R. E. Lopez-Herrejon, M. Schulman, "Using Interactive Technology in a Short Java Course: An Experience Report," ACM SIGCSE Bulletin, Volume 36, Issue 3, September 2004.
- [12] J. H. McMillan, Classroom Assessment: Principles and Practice for Effective Instruction, Third Edition, Allyn & Bacon publishing, 2003.
- [13] S. Ritter, J. Anderson, M. Cytrynowicz, and O. Medvedeva, Authoring Content in the PAT Algebra Tutor, (1998) Authoring Content in the PAT Algebra Tutor. Journal of Interactive Media in Education, 98 (9) [www.jime.open.ac.uk/98/9/], Published 8 Oct., 1998.
- [14] F. Fraij, "Visual and Interactive Computer-Based Software for Teaching a Mathematics Course: An Approach", Proceedings of the IEEE Conference on AI Tools in Engineering, Pune, India, 2008.
- [15] F. Fraij and I. Al-awamleh, "SFRL: Intelligent, Web-Based Tutoring Software for Teaching a Mathematics Course" International Journal of Computing and Information Sciences, submitted for publication.
- [16] D. Dagger, V. P. Wade, and O. Conlan, Developing Active Learning Experiences for Adaptive Personalised eLearning, Third International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH2004) Proceedings, Eindhoven, The Netherlands (2004)
- [17] Microsoft.NET Framework: www.microsoft.com/NET/
- [18] VisualBasic.NET: msdn.microsoft.com/en-us/vbasic/
- [19] H. Sharp, Y. Rogers, and J. Preece, Interaction Design beyond Human-Computer Interaction, Wiley, 2007, 978-0-470-01866-8
- [20] D. Rogers, D. Lambert, and A. Knemeyer, "The Product Development and Commercialisation Process", The International Journal of Logistics Management, Vol. 15, No.1, 2004.