An Intelligent Interactive Tutoring System For Engineering Education

Haresh Shahani, Saroj K. Biswas, Musoke Sendaula, Harish Pallila

Department of Electrical and Computer Engineering
Temple University
Philadelphia, PA19122
{haresh, sbiswas, musoke, pallila}@temple.edu

Abstract

We present the architecture of an Intelligent Interactive Tutoring System (IITS) that is web-based, and can be adapted to courses in engineering, sciences, and mathematics. This research seeks an architecture that can be used by an instructor to make an online learning system for a course the instructor wants to teach in classroom or online, rather than developing an in-depth tutoring system for any specific course. The Intelligent Interactive Tutoring System Shell integrates mathematical tools and an expert-system-type logical analysis/synthesis tool in a web-based environment. The IITS consists of several components including an instructor interface, a student interface, a student model, a student log, a reasoning system, and a mathematical tool interface module, and guides the student through a monitored problem solving session.

1. Introduction

Students need academic support outside the classroom, which may be provided by a professor after the regular class hours, or by a tutor, between certain fixed hours of the day. However, there are many students who are hard pressed between work and study, shuttling between the workplace to earn a livelihood, and the college in preparation for a better livelihood. As such, it is not possible for these students to get appropriate academic help during regular office hours, nor is it possible for them to see a tutor at an odd hour when they return home from work. A possible approach to providing academic support to these students is the use of an intelligent tutor that mimics the functions of a human tutor.

Over the last few years, various on-line learning systems and tutoring systems have been developed across the globe. This includes commercial knowledge management systems, such as, Blackboard [1], and WebCT [2], as well as the distance education systems developed at various academic institutions [3-14]. A majority of e-Learning systems concentrate mainly on delivery of course contents over the Internet with little or no room for interactivity. Interactivity
in the form of live discussions, chat, or through whiteboard applications has been introduced in some cases, which however require that the instructor or fellow students remain online. Web based interactive laboratory experiments have also been developed for electrical circuits, electronics, fluid mechanics, and control systems courses.

The primary focus of this paper is the development of an interactive tutoring system that has some level of intelligence as a human tutor. A human tutor will try to determine what deficiencies a student might have, will keep track of when the student stumbles in solving a problem, and will suggest additional support material for review. In addition, the tutor is also likely to coordinate activities with the class instructor, and will send the instructor necessary feedback about the student’s progress.

With support from the National Science Foundation, a prototype of the Intelligent Interactive Tutoring System (IITS) shell has been developed. It is a web-based environment that integrates the domain knowledge on the subject area with numerical and engineering analysis tools, logical analysis tools, in an interactive user interface. It consists of three primary modules: a) domain knowledge module, b) student model module, and c) mathematical tools module, and interfaces with the users through the d) student interface module, and the e) instructor interface module, and reports to the instructor about student’s progress through the f) student log module.

The IITS is a generic shell that can be adapted to most courses in engineering and sciences, rather than being a detailed tutoring system for a particular course. The instructor provides the domain knowledge in various formats, such as, text, audio, video files, and animations. The instructor also serves as the ‘expert’ for the logical analysis system that guides the student through problem solving sessions. The IITS builds a student model for the student’s academic preparation and progress based on activities during problem solving sessions, and guides the student through the problem, as well as suggests background material to the student as required. The IITS also reports to the instructor about the student’s progress in a course.

The architecture of the Intelligent Interactive Tutoring System is presented in Section 3. Monitored problem solving session using IITS is discussed in Section 4, which is followed by some concluding remarks in Section 5.

2. e-Tutoring Techniques

There are many students who work over 20 hours per week to cover living expenses and tuition. Although many schools routinely offer tutoring support, a vast majority of students are unable to come to the tutoring centers that are typically open during regular office hours. Thus it is necessary to develop a system that mimics the teaching style of a human tutor. The e-tutoring system must be capable of guiding the student when he or she stumbles on a problem, suggest background or prerequisite material, understand the student’s readiness for the course, and keep track of student’s progress through the course.
The functions of a tutor can be broadly classified into several categories: a) convey the concepts using static media, b) convey the concepts using dynamic media, c) monitored problem solving, d) guidance for prerequisites and backgrounds, and e) monitor progress through the course.

The concepts that are to be taught are presented to the student in the form of textual explanations, descriptive figures and examples. With advances in computer technology, it is also possible to use multimedia technologies, such as, video, audio, and interactive simulations to promote the student’s understanding of the particular concept. Interactive simulations foster learning by supporting the development of accurate mental models. A mental model, as defined by cognitive scientists, is “an internal representation of a physical system used by a person or a program to reason about processes involving that system” [15]. In engineering fields, concepts are better learned through an interactive simulation of the process. For example, a typical simulation would be that of a series-parallel resistive circuit with an input voltage. The interactive simulation could allow the student to vary the source voltage, and different resistances in the circuit, and observe the changes in the voltage or current. As the student varies the resistor values, changes in the circuit properties can be made visible through changes in a text file as well as through graphics. This contributes significantly to the student’s understanding of circuit concepts.

e-Learning systems are particularly suitable to convey the concept through multimedia tools including audio/video, and interactive simulations. Various audio, video, and text files are easily transmitted to the students through Blackboard [1] and webCT [2] environment as well as non-commercial e-Learning systems developed at various universities recently. These systems also allow certain level of interactivity through the use of whiteboard, chat, email, and other interfaces. What is lacking in most e-Learning systems is an environment for monitored problem solving. In a monitored problem solving environment, the student can be prompted with background material should he/she stumbles at a certain step during solving a problem, will guide the student to take an action leading to successful completion of the solution, and will keep track of the student’s progress over time. This research develops an architecture for a web-based e-Tutoring system with the primary focus on monitored problem solving. In what follows we present the details of the Intelligent Interactive Tutoring System.

3. Architecture of Intelligent Interactive Tutoring System

The Intelligent Interactive Tutoring System integrates typical numerical tools, such as, Matlab, and logical analysis/synthesis tools, such as, expert systems, in an interactive web-based environment. It consists of three primary modules: a) domain knowledge module, b) student model module, and c) mathematical tools module, and interfaces with the users through the d) student interface module, and the e) instructor interface module, and reports to the instructor about student’s progress through the f) student log module. The IITS system is written in web programming language java, and uses the MySQL database.
3.1 The Domain Knowledge Module

The Domain Knowledge Module, as the name suggests, stores knowledge about the subject domain that is to be communicated to the student. Domain knowledge consists of problem solving expertise, explanations, examples, and interactive simulations, and arranged in a hierarchical structure, called the ‘domain knowledge graph’. The domain knowledge could be stored in the form of audio, video, and text files in various formats.

3.1.1 Domain Knowledge Graph

The concepts that make up the subject are represented as distinct topics that are to be taught to the student. These topics are stored in a ‘domain knowledge graph’ to capture the progress through the subject. For example, in circuit theory, Ohm’s law is recorded as a pre-requisite for the Kirchoff’s voltage law in the graph. This provides the system with an order in which to present the material to the student. This information is also used by the system to guide students when they interact with the system.

3.1.2 Explanations, Examples, Interactive simulations

The topics in the domain knowledge graph have textual explanations, examples, and problems associated with them. Explanations elucidate theoretical concepts and examples demonstrate their applications. Interactive simulations, in the form of Java applets, allow the student to play around with standard experimental setups and learn interactively by observing changes in the output parameters. Various file formats (audio, video) are supported for the explanations and examples. The course construction interface allows the instructor to build the domain knowledge graph and attach explanations, examples and interactive simulations to specific concepts.
Figure 2: Tutorial Files provide explanations and examples

Figure 3: Interactive Simulation for the concept of KVL

“Proceedings of the 2004 American Society for Engineering Education Annual Conference & Exposition
Copyright © 2004, American Society for Engineering Education”
3.1.3 Problem solving Expertise

Each problem is associated with a set of methods for solution. A solution to a problem is represented as an application of several smaller ‘steps’ applied in a sequential manner. The steps that make up solutions are organized into a database. For each step, the database stores the following:

1) The logic used to solve the step (in the form of a java program)
2) A link to the concept in the domain knowledge graph
3) A link to the domain knowledge graph on prerequisite material
4) Explanations that provide help and hint for the particular step.
5) Examples for the particular step.
6) The logic that is used to compare the student’s answer to the correct answer (in the form of a java class, which is called the comparator module).

Database of Steps

- Logic for the step (java program)
- Link to a concept
- Explanation that provides hint
- Similar examples of the step
- Comparator module (checks student answer)

Figure 4: Database of steps

The above scheme of breaking down solutions into smaller steps, and maintaining expertise for each step provides a systematic method for building problem solving expertise, and facilitates monitoring of student’s progress. This module is essentially the reasoning system component of the Intelligent Interactive Tutoring System that provides problem-solving support to the student, identifies the mistakes done by a student, and guides the student to correct mistakes.

3.2 The Student Model

The student model collects diagnostic information about the student’s preparation and progress through its interactions with the student. Ideally, this model should include all the aspects of the student’s behavior and knowledge that have repercussions for his performance and learning. This means that successful completion of a particular step means the student has satisfactorily understood the application of the particular concept of the domain. Likewise, when the student makes a mistake in a particular step, his or her weakness of that particular concept is recorded. Thus, the student model contains information about the student’s strengths and weaknesses with respect to concepts in the domain. This information is used to customize the system’s interaction.
with the student. After each problem is solved, the student model is checked to see whether the student’s performance in a particular concept has gone below a threshold value. If a student shows lack of background for a certain step of a problem, he/she is advised to go through the relevant concept and its prerequisites.

3.3 The Student Log

The student log records all interactions the student makes with the system. However, the information in the student log serves a different purpose, and hence, is of a different form as compared to the student model. The student log serves as a source of feedback of the student’s usage of the system for the teacher including student’s strengths and weaknesses. The teacher can use the student log to find out how the student is progressing through the course, and also how the class fared while solving a particular problem or a group of problems.

3.4 The Student Interface

The student interacts with the system using the student interface. This component handles student requests for explanatory material, and presents it in an easy-to-use web-based environment. It is also responsible for providing the interactive web-based problem-solving interface to the student. The student model is used by the student interface to customize its interaction with the student. The student interface and the student model share a circular relationship, i.e., are dependent on each other. The student model is populated by the student interface, which, in turn, uses the same information stored in the student model to guide future interactions with the student.

Figure 5: The Student Interface
3.5 The Teacher/Instructor Interface

The instructor interface is the instructor’s view of the system, and allows the instructor to present the course material using text, audio, graphics, video, and other data formats. This may include recorded classroom lectures, supplemental materials, sample questions, audio clips, short movie clips, sketches done on sketchpad, etc.

Building the problem solving expertise is one of the major functions of the Course Construction Interface, and is achieved by building the 'Database of Steps', followed by using the steps defined to build solutions to problems. The interface has methods that allow the instructor to add new steps to the database, modify existing ones and delete those that are not needed. Each step also has a link to a concept in the domain knowledge graph, explanations, examples and a comparator module associated with it.

3.5.1 The Course Construction Interface

The Course Construction interface is used to build the domain knowledge in the form of explanations, examples, interactive simulations, and problem solving expertise. It allows the teacher to build the course contents on-line in a systematic and modular manner. Details for building the problem solving expertise are handled by the course construction interface. The instructor can also build exercises for the students to solve by choosing problems stored in the system.

Figure 6: The Course Construction Interface
3.5.2 The Course Monitoring Interface

The teacher uses the Course Monitoring interface to retrieve the information in the student log. It serves as a feedback about the performance of different students as they use the system. The Course Monitoring Interface is used to view the performance of students as they solve exercises and work through problems.

3.6 The Mathematical Tools Interface

The system allows the use of existing mathematical tools (e.g. Mathematica, Matlab) to support the problem solving expertise. The mathematical tools interface is used by the course construction component in this context. Its main function is to provide a simple interface for building the problem-solving expertise.

3.7 Implementation

The system has been developed using the Java programming language. JSP technology is used to make the system web-based and dynamic. The object-oriented capabilities of Java have helped to develop the system in a systematic and modular manner. Each module shown in the architecture is represented by a different piece of Java code, and these pieces of Java code interact and communicate with each other to achieve the functionality of the system.

4. Monitored problem solving using IITS

Problem solving is an important activity in any mathematics, science or engineering related curriculum. Concepts learnt in theory are applied during problem solving. The IITS system guides the students during this important activity, and uses it to diagnose the student’s weaknesses and strengths.

The student has the opportunity to solve problems after he or she has gone through the theory related to a particular topic. For a particular problem, the student has the opportunity to choose a particular method from the list of methods. As the student chooses a particular method, the problem is broken down into ‘steps’, and presented to the student. The student can then solve the problem step-by-step under the guidance of the system. The system makes sure that the student produces correct results at the end of each step. For each step of the problem, hints and explanatory figures are available that the student may review. The student can even choose to view additional worked-out examples that elucidate a particular step. In this manner, the student has access to relevant help material as and when needed.

In the background, as the student is working through the problems, the ‘student model’ is populated by the system. Problems and steps that make up their solutions are organized according to the concepts they stand for. As the student makes mistakes and stumbles on particular steps, the system records that information to indicate weaknesses in understanding particular concepts.

The system uses the information in the student model, and develops a custom exercise for the student to complete. The student model contains information about the student’s level of
understanding of the domain. The system builds up the custom exercise by choosing those problems that would improve the concepts in which weakness has been demonstrated. Apart from using the diagnostic information in the student model to develop practice exercises, the system builds a set of directives for the student. These set of directives serve as feedback to the student, and suggest topics that he/she should consider reviewing again in greater detail.

Figure 7: The system checks the answer entered by the student and provides help.

5. Conclusions

Online tutoring that is interactive and adaptable to the student’s needs goes a long way in promoting understanding of key concepts and their applications. Interactive simulations provide a rich learning environment where students can gain a thorough understanding of the simulated systems by observing the results of experiments they carry out.

This paper presents an Intelligent Interactive Tutoring System (IITS) that provides a monitored problem-solving environment to the student. The IITS is a web-based environment that is customizable for courses in science and engineering. It integrates domain knowledge on the subject area, a mathematical engine, a reasoning system, and a student model with appropriate student interface and instructor interface. The use of a student model makes it possible to provide the student help when it is needed and where it is needed.
6. References


2. WebCT Inc., http://www.webct.com


HARESH SHAHANI
Haresh Shahani is a graduate student in the Department of Computer and Information Sciences, Temple University. He completed his undergraduate degree in Computer Engineering from University of Mumbai in 2002. His primary area of research is artificial intelligence.

SAROJ BISWAS, Ph.D.
Saroj Biswas is an associate professor on Electrical and Computer Engineering, Temple University, Philadelphia since 1986. His primary areas of research are control systems, artificial intelligence, neural networks.

MUSOKE SENDAULA, Ph.D., P.E.
Musoke Sendaula is a Professor in the Department of Electrical and Computer Engineering, Temple University since 1986. He has an extensive research record in power systems and control systems, and applications of intelligent systems.

HARISH PALLILA
Harish Pallila is a graduate student in the Department of Electrical and Computer Engineering, Temple University. He graduated in Electrical Engineering from Birla Institute of Technology and Science in 2002. Currently he is working on intelligent systems towards his master’s degree.