2006-2209: AN INTELLIGENT INTERACTIVE TUTORING SYSTEM FOR AN ELECTRIC CIRCUITS COURSE

Saroj Biswas, Temple University

Saroj Biswas is a professor of Electrical and Computer Engineering, Temple University, Philadelphia. He received his doctoral degree in Electrical Engineering from University of Ottawa, Canada, in 1985. His primary areas of research are control systems, artificial intelligence, and neural networks. He is a member of IEEE, ASEE, and Sigma Xi.

Musoke Sendaula, Temple University

Dr. Musoke Sendaula is a Professor of Electrical and Computer Engineering, Temple University. He received his master's and doctoral degrees from the University of Conneecticut in 1970 and 1972, respectively. His research interests include intelligent systems, computer communication and security, and power systems. He is the coauthor of a text book on power quality control, and many publications. He is a member of IEEE, Tau Beta Pi, Eta Kappa Nu, Phi Kappa Pi, and Sigma-Xi.

Sesha Yeruva, Temple University

Sesha Yeruva completed her master's degree in Electrical Engineering from Temple University in 2005. She is currently with Siemens Medical Inc as a Circuit Design Engineer.

Krishana Priya Sannidhi, Temple University

Krishna Priya Sannidhi is a graduate student in Electrical Engineering, Temple University. Her primary area of researh is digital signal processing.

Ravi Shankar Dwivedula, Temple University

Ravi Shankar Dwivedula is a graduate student in Electrical Engineering, Temple University. His primary area of interest is signal processing.

An Intelligent Interactive Tutoring System for an Electric Circuits Course

Abstract

This paper summarizes the development of an intelligent tutoring system for a sophomore level course sequence on electric circuits. The tutoring system is based on the Intelligent Interactive Tutoring System Shell, developed earlier. It has several features: a) it is web based which makes it accessible to students at any time, b) it is interactive much like the way a student interacts with a human tutor, c) it is intelligent in the sense that it mimics the teaching style of a human tutor, and d) it is adaptable by any instructor. The course material is organized as a collection of logical interrelated topics with examples, and exercise problems. The tutoring system maintains a database for every interaction that the student makes with the system. As the student progresses through a course, the system records every move that the student makes, such as, the number of attempts that the student makes in solving a particular problem, whether the student is able to solve a problem, etc., and suggests prerequisites as the student stumbles on a problem. This tutoring system is flexible enough to allow any instructor to configure it for his or her own course.

1. Introduction

With advances in computer technology, it is possible to use multimedia technologies, such as, video, audio, and interactive simulations to promote the student's understanding of the particular concept. Such computer based multimedia technologies could be broadly classified into two categories: e-learning systems, and e-tutoring systems.

e-Learning systems are particularly suitable to convey the course material 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[3, 4, 5] that have been developed at various academic institutions. Web based interactive laboratory experiments have also been developed for electrical circuits[5,6], electronics[7,8], fluid mechanics, and control systems courses[9, 10, 11, 12], and various other courses. Some of these systems also allow certain level of interactivity through the use of whiteboard, chat, email, and other interfaces, however, lack an interactive environment for monitored problem solving, and feedback. On the other hand, an e-tutoring system is ideally an electronic counterpart 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 basic architecture of a web-based e-Tutoring system, Intelligent Interactive Tutoring System (IITS) Shell, has been developed in[15,16]. The IITS has several features: a) it is web based which makes it accessible to students at any time, b) it is interactive much like the way a student interacts with a human tutor, c) it is intelligent in the sense that it mimics the teaching style of a human tutor, and d) it is adaptable by any instructor. The IITS is an open-ended shell structure that can be used by an instructor to develop an interactive tutoring system using the instructor's own course material in audio, video, text, simulation, and other forms. It is a generic shell that can be adapted for most courses in sciences and engineering. Besides being a web-based environment for content delivery of the course material, the IITS also provides an environment for 'monitored-problem-solving' that a human tutor offers to a student.

This paper presents the adaptation of IITS for the sophomore circuit course sequence, EE161: Electrical Engineering Science I, and EE165: Electrical Engineering Science II, for Electrical Engineering students. The first course is the basic dc circuits course, while the second course EE165, is an extension into the AC circuits.

The paper is organized as follows: Section 2 is an overview of the IITS shell that has been reported earlier[15,16]. Section 3 presents the development of the two courses in the IITS shell structure. Various modules for the two courses, and screen shots of student and instructor interfaces are discussed. Section 4 is a description of Mentored Problem Solving environment that guides a student through a problem solving session. Section 5 has some concluding remarks.

This paper is more of a progress report rather than a full description of a matured version of the two courses mentioned earlier. Various modules are still being developed, and a classroom evaluation of the tool/courses is yet to be done. The authors hope that development of the various modules will be relatively more complete by the end of the Spring semester, just in time to present a better report to the conference.

2. *IITS:* Intelligent Interactive Tutoring System Shell

For the sake of completeness of presentation, we first give a brief summary of the IITS architecture. The IITS shell is an integration of various mathematical tools and an expert-system-type logical analysis/synthesis tools in a web-based environment. The IITS consists of six primary modules:

1) Domain knowledge module
2) Student model module
3) Student interface module
4) Instructor interface module
5) Student log module
6) Mathematical tools module

The IITS system is written in web programming language Java, and uses the MySQL database. 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.

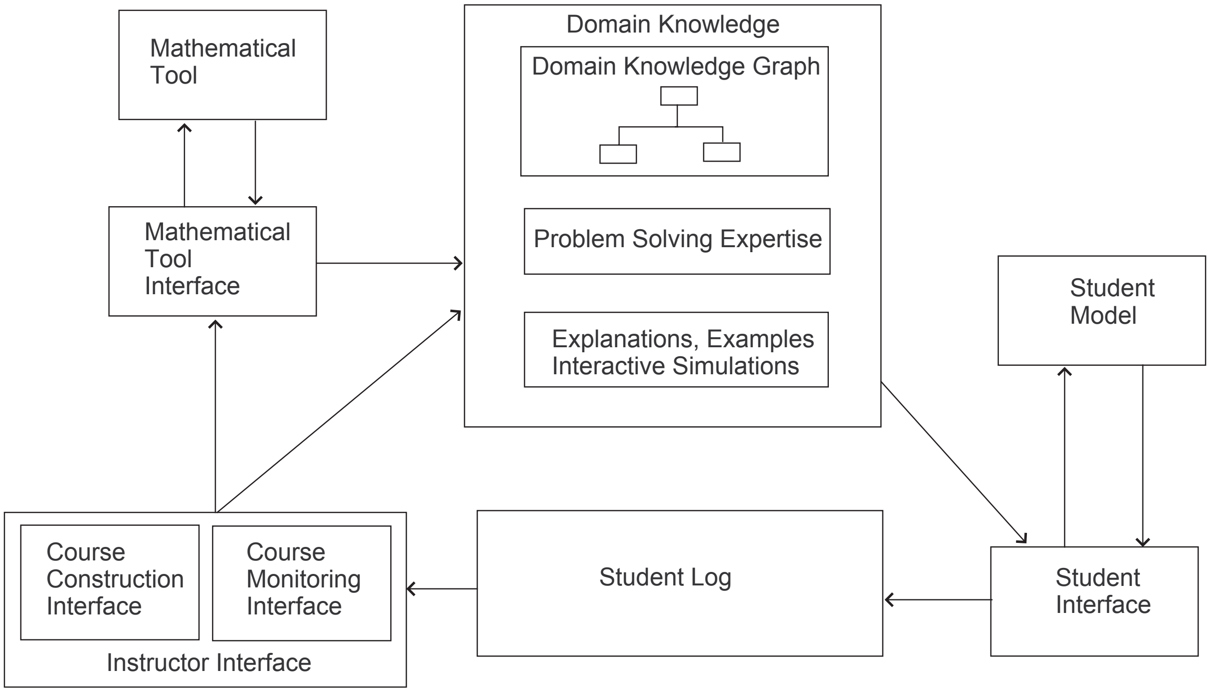

Mathematical Tool
Domain Knowledge
Domain Knowledge Graph
Mathematical Tool Interface
Problem Solving Expertise
Student Model
Explanations, Examples Interactive Simulations
Course Construction Interface
Course Monitoring Interface
Student Log
Student Interface
Instructor Interface

Fig 1 IITS Architecture

Domain knowledge module: The Domain Knowledge Module stores knowledge about the subject domain that is to be communicated to the student. It consists of problem solving expertise, explanations, examples, interactive simulations, and various course-specific materials 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.

The Student Model: The student model collects diagnostic information about the student's preparation and progress through its interactions with the student. The model includes all aspects of the student's behavior and knowledge that have repercussions for his/her performance and learning. 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.

The Student Interface: The student interacts with the system using the student interface. This component is 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 model is populated by the student interface, which, in turn, uses the same information to guide future interactions with the student.

The 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. The instructor interface handles both course construction and course monitoring. The Course Construction interface is used to build the domain knowledge in the form of explanations, examples, interactive simulations, and problem solving expertise. The instructor interface also contains a Course Monitoring interface to retrieve the information about students' progress through a course, either for each individual or as a group.

Student Log Module: The student log records all interactions the student makes with the system including number of attempts and ability to solving a problem. The student log serves as a source of feedback of the student's usage of the system for the instructor including student's strengths and weaknesses. The instructor 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.

Mathematical Tools Interface Module: Use of mathematical tools, such as, Matlab, is common in engineering education. Although a full integration of Matlab core into IITS has been the goal of the project, currently, student's answers for exercise problems are verified using a java-based small linear algebra module.

3. Development of Interactive Tutoring System for ECE Circuits Courses

Using the IITS shell, an interactive tutoring system is being developed for two electric circuits courses for the Electrical Engineering curriculum. Several modules for various components of the electric circuits courses have been developed and deployed; further development is in progress. The following is an index of various topics covered so far:

3.1 Course Topics

EE161: Electrical Engineering Science I
Electric circuit elements – help files
Branches & nodes – help files
Complex algebra – simulation module
Kirchoff's current law – help files, step-wise help file, simulation module
Kirchoff's voltage law – help files, step-wise help file, simulation module
Series resistance & parallel resistance – help files, step-wise help file
Mesh & nodal analysis – help files, problem files
Source transformation – help files, problem files, simulation module
Superposition – help files, problem files, simulation module
Thevenin's theorem – help files, problem files, step-wise help file, simulation module
Norton's theorem – help files, problem files, step-wise help file, simulation module

EE165: Electrical Engineering Science II
Energy Storage Elements – help files, examples
First Order Circuits – help files, examples
Second Order Circuits – help files, examples
Phasors and AC Circuit Analysis – help files, examples
Power and Reactive Power – in progress
Transformers – in progress
Frequency Response – in progress
Laplace Transform and its applications – in progress

3.2 Course Material

The course covers the basic and advanced concepts of electric circuits as two semester sequence. Basic material for these two courses can be found in standard textbooks on electric circuits. This section essentially describes how the course material is presented and organized in the IITS format. The four main ways of explaining the topics in the course are: using the help files (.pdf, and .swf), problem files (.pdf), interactive simulation modules (.java), and step-wise help files (.java).

Help files:
Help files are organized in two ways: a printable file in .pdf format, and a movie file in .swf format. Most of the help files contain a problem on the topic being discussed and the method of solving the problem. Students may read the help files from the .pdf file, or listen to the movie.

Problem files:
Problem files contain several problems on each topic of the course. These files should make the student familiar with the topics by covering a large set of problems with different levels of difficulty (easy, medium, difficult, and very difficult).

Simulation modules:
Interactive simulation modules have been designed to give better insight on some of the important topics in electric circuits. These java modules generally contain a problem on the topic being discussed and interactively generate a solution to the problem, as the parameters or values of circuit elements are being changed. For example, students could vary the resistance, inductance and capacitance values in a circuit, and observe the corresponding changes in the KCL equation, and the mesh currents.

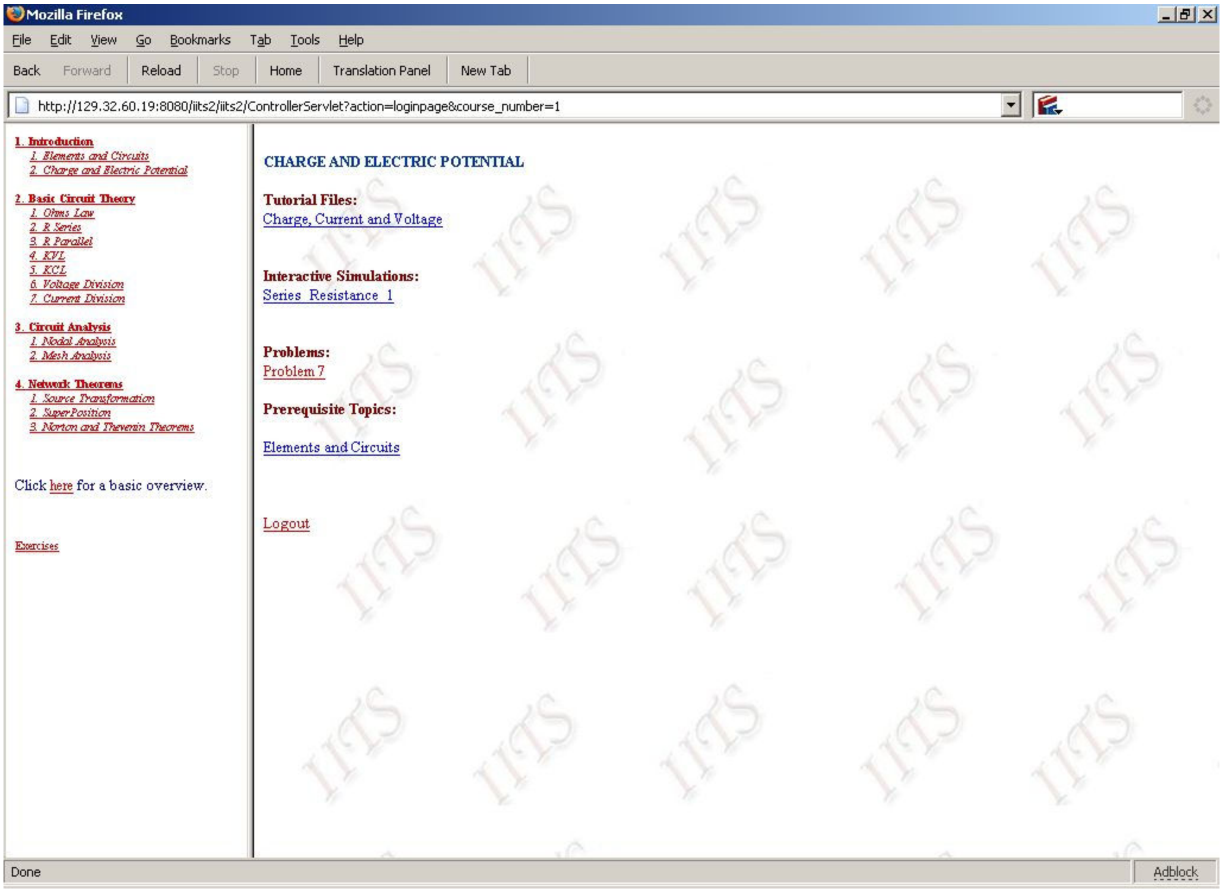
Step-wise help files:
These java files also contain problems on several topics of electric circuits, but unlike simulation modules, generate a step-wise method of solving a problem. These modules solve a problem step-by-step, and prompt the student to enter intermediate results or equations at each step. The answer entered is then compared with the result obtained by the module itself. If the student's answer is correct, he is then encouraged to proceed to the next step; otherwise, the module prompts the student to check the help files related to the topic.

Problem database:

A set of supplementary problems on each topic are stored in the problem database. These problems, as various levels of difficulties, are prompted to the student as he/she goes through a particular topic.

3.3 Student Interface

The student logs in using a system generated username and password. The student also selects one of the two courses to study; EE161 or EE165. The system will then show a welcome screen with a course subtopics listed in a panel on the left. This should look similar to the following image shown in Figure 2.


Mozilla Firefox
File Edit View Go Bookmarks Tab Tools Help
Back Forward Reload Stop Home Translation Panel New Tab
Done
Adblock

http://129.32.60.19:8080/iits2/iits2/ControllerServlet?action=loginpage&course_number=1

1. Introduction
- 1. Elements and Circuits
- 2. Charge and Electric Potential

2. Basic Circuit Theory
- 1. Ohms Law
- 2. R Series
- 3. R Parallel
- 4. KVL
- 5. KCL
- 6. Voltage Division
- 7. Current Division

3. Circuit Analysis
- 1. Nodal Analysis
- 2. Mesh Analysis

4. Network Theorems
- 1. Source Transformation
- 2. SuperPosition
- 3. Norton and Thevenin Theorems

Click here for a basic overview.

Exercises

CHARGE AND ELECTRIC POTENTIAL

Tutorial Files:
Charge, Current and Voltage

Interactive Simulations:
Series Resistance 1

Problems:
Problem 7

Prerequisite Topics:

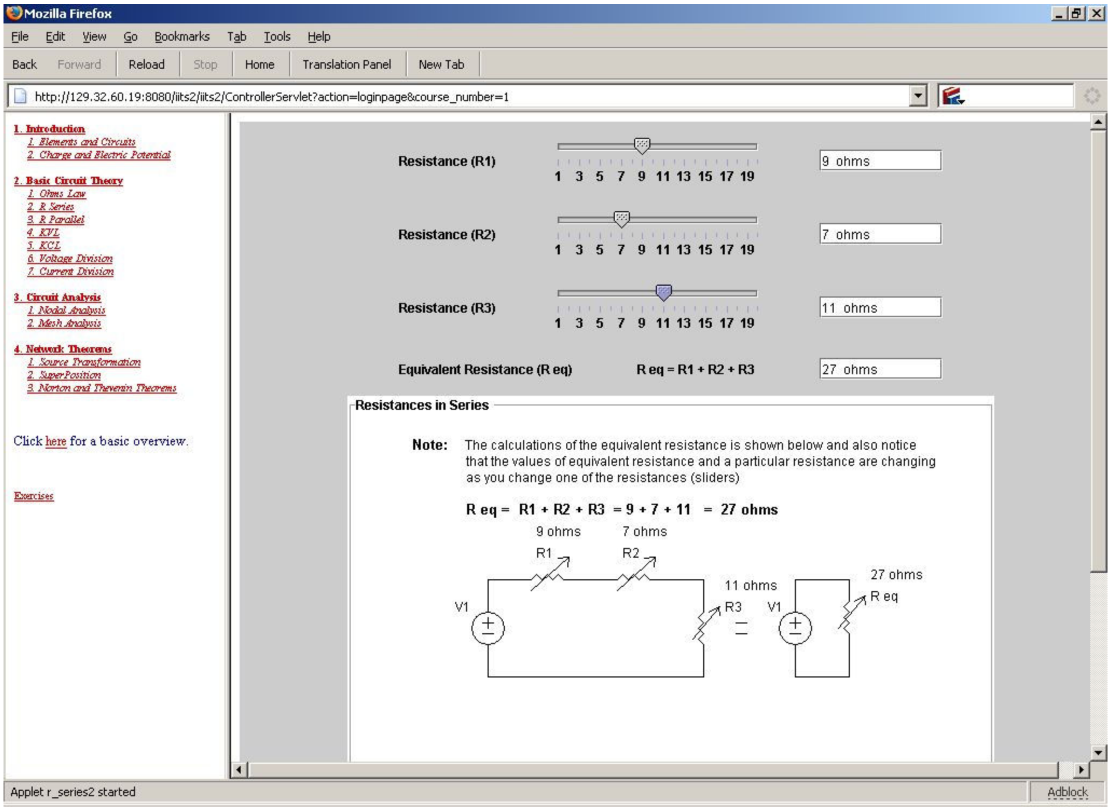
Elements and Circuits

Logout

Fig 2 Screen Shot of Student Interface

The panel on the left enables the student to select one of the several subtopics for the course, which are essentially various chapters in standard textbooks. The student then selects one of the topics to study, and the corresponding components, such as, tutorial files, interactive simulation files, prerequisite files, problem files, etc.

Here is an example of '**Interactive Simulations**', subtopic '**Series Resistance 1**'. Using the sliders, students can change various resistances in the circuit, and see changes in the circuit equation, and the mesh current. Figure 3 shows the screen.



Resistance (R1) 9 ohms

Resistance (R2) 7 ohms

Resistance (R3) 11 ohms

Equivalent Resistance (R eq) R eq = R1 + R2 + R3 27 ohms

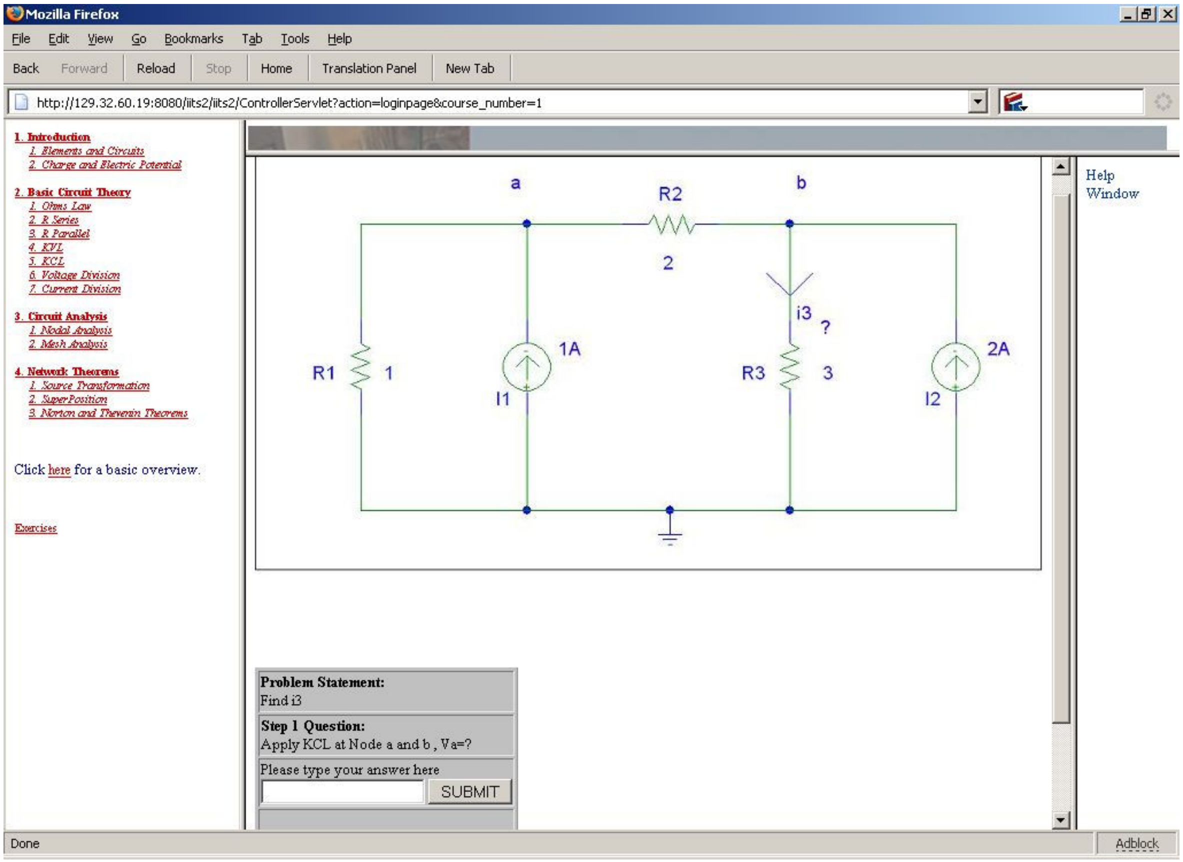
Resistances in Series

Note: The calculations of the equivalent resistance is shown below and also notice that the values of equivalent resistance and a particular resistance are changing as you change one of the resistances (sliders)

R eq = R1 + R2 + R3 = 9 + 7 + 11 = 27 ohms

Fig 3 Interactive Simulation

Now let us, for instance, navigate to the subtopic **Nodal Analysis**. Under the section '**Problems**', click on the link '**Problem 19**'. A problem with a circuit diagram and the question under it would appear. The problem would also show the step number and the question to be solved for that step. A text-box is also provided for the answer. The screen should look similar to the following image in Figure 4.


Mozilla Firefox
File Edit View Go Bookmarks Tab Tools Help
Back Forward Reload Stop Home Translation Panel New Tab
http://129.32.60.19:8080/iits2/iits2/ControllerServlet?action=loginpage&course_number=1
a
R2
b
2
i3
?
1A
2A
R1
1
R3
3
I1
I2
Help Window
Done
Adblock

1. Introduction
   1. Elements and Circuits
   2. Charge and Electric Potential

2. Basic Circuit Theory
   1. Ohms Law
   2. R Series
   3. R Parallel
   4. KVL
   5. KCL
   6. Voltage Division
   7. Current Division

3. Circuit Analysis
   1. Nodal Analysis
   2. Mesh Analysis

4. Network Theorems
   1. Source Transformation
   2. SuperPosition
   3. Norton and Thevenin Theorems

Click here for a basic overview.

Exercises

Problem Statement:
Find i3

Step 1 Question:
Apply KCL at Node a and b , Va=?

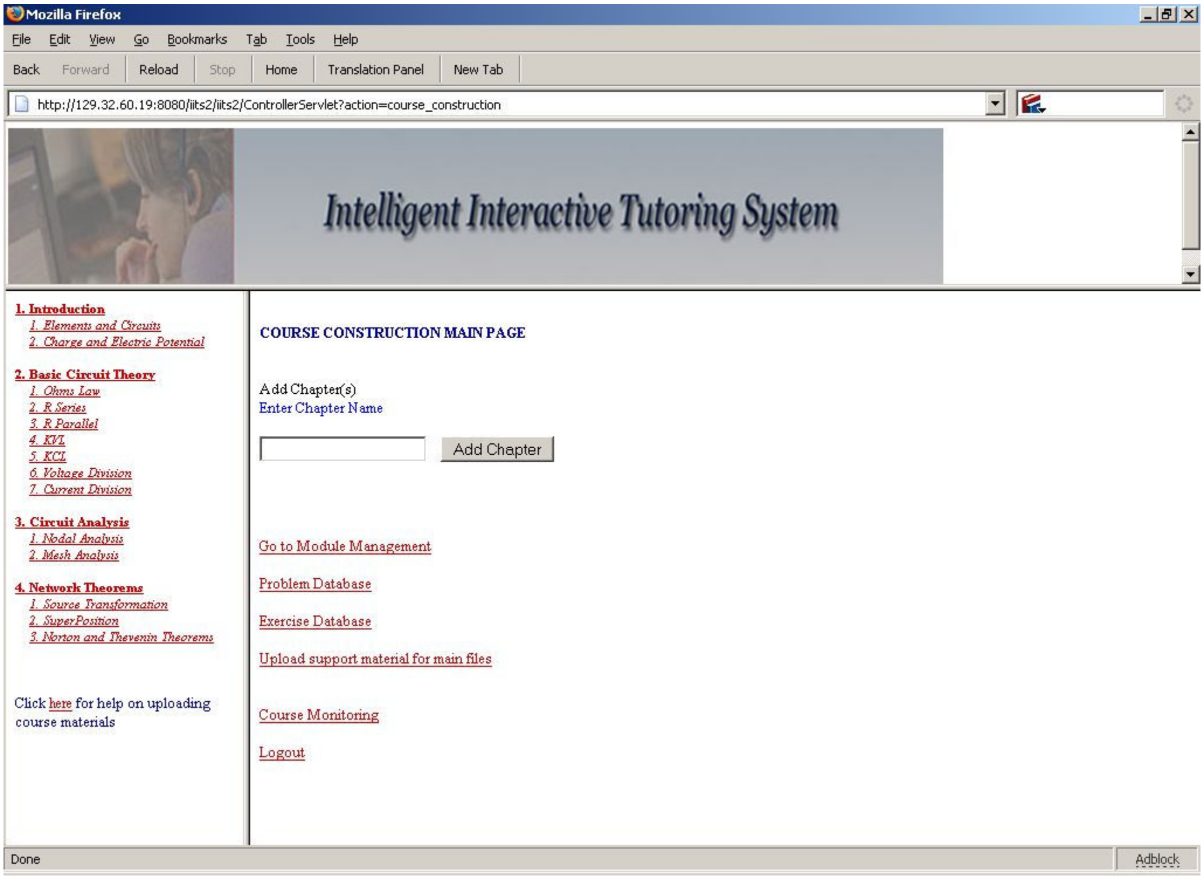
Please type your answer here
SUBMIT

Fig 4 Step-wise solution environment for Circuit Problems

3.4 Instructor Interface

The instructor interface is divided into two parts: course construction interface, and course monitoring interface. The instructor chooses one of the two interfaces after a standard login procedure.

The course construction interface allows the instructor to develop a new course and/or modify an existing course. Functionalities included in this interface are: a) add a new chapter, b) add sections for a chapter, and c) upload files. The files could be uploaded from the instructor's computer in any one of the supported formats. There is also an option for inserting the document using a textbox. For each chapter, the instructor navigates to one of the databases: problem database (for worked out examples), exercise database (for problems for the students to work on), help databases, etc, and uploads the files as necessary. The interface is intuitive, however, a detailed *User's Manuel* has been developed, and is available online.


Mozilla Firefox
File Edit View Go Bookmarks Tab Tools Help
Back Forward Reload Stop Home Translation Panel New Tab
http://129.32.60.19:8080/iits2/iits2/ControllerServlet?action=course_construction
Done
Adblock

Intelligent Interactive Tutoring System

1. Introduction
1. Elements and Circuits
2. Charge and Electric Potential

2. Basic Circuit Theory
1. Ohms Law
2. R Series
3. R Parallel
4. KVL
5. KCL
6. Voltage Division
7. Current Division

3. Circuit Analysis
1. Nodal Analysis
2. Mesh Analysis

4. Network Theorems
1. Source Transformation
2. SuperPosition
3. Norton and Thevenin Theorems

Click here for help on uploading course materials

COURSE CONSTRUCTION MAIN PAGE

Add Chapter(s)
Enter Chapter Name

Add Chapter

Go to Module Management

Problem Database

Exercise Database

Upload support material for main files

Course Monitoring

Logout

Fig 5 Instructor Interface

For the course monitoring interface, the instructor reviews the student's performance for the course. In particular, the instructor could see how many times the student has attempted a problem, and whether it has been successfully completed. The instructor could also compare the student's performance with the rest of the class.

3.5 Help for the Instructor for Problem Generation

A module to solve an electric circuit using mesh and nodal analysis has been developed to assist the instructor develop problems for the course. This module is a PSpice-type environment in java to solve circuit problems using mesh or node analysis methods. The module creates a 'netlist' of the circuit much like the way PSpice creates a netlist for electric circuit problems. Circuit equations are developed using this netlist, and solved using standard methods for solving matrix equation. The system is capable of handling complex numbers so that one can solve circuits containing inductors and capacitors as well. This module is not available to the students.

The netlist is saved in the following format:

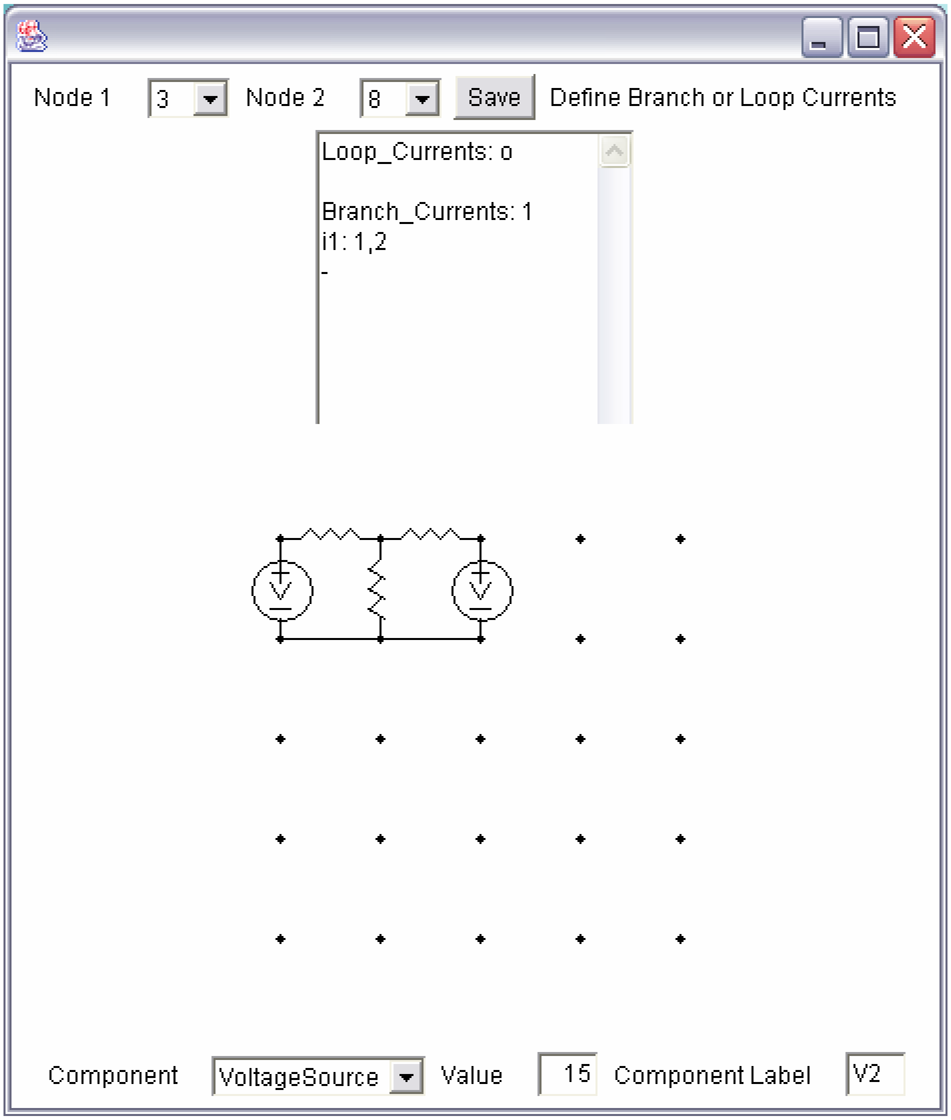
All resistors/voltage and current sources are saved sequentially as

```
R = [ node1 node2 value_R1;
     node1 node2 value_R2;
     ...   ...   ...];
V = [ node1 node2 voltage_V1;
     node1 node2 voltage_V2;
      ...   ...   ...];

Is = [ node1 node2 current_I1;
     node1 node2 current_I2;
     ...   ...   ...];
```

The netlist is generated automatically as the instructor ‘draws’ a circuit on the screen using ‘Problem Generator’ module, which is shown in the next figure. Problem generator imitates PSPICE to draw any electric circuit on-the-fly using simple buttons. It contains a matrix of nodes, and the user can draw the components like resistor, voltmeter, ammeter, wire, ground etc. between any two consecutive labeled nodes, horizontally or vertically.

Once the circuit is ‘drawn’ on the screen, a ‘netlist’ is created for all circuit components, their connectivity between various nodes, and their values. Another java modules then decomposes the ‘netlist’ in terms of ‘node incident matrix’ and ‘component matrix’, which is essentially a matrix representation of the circuit, and finds its solution. This java module also computes currents and voltages for all elements in the circuit. This module gives the instructor an easy way to generate new problems, which can be presented either as “worked out examples” or as “exercise problems” for the students.



Node 1 3 Node 2 8 Save Define Branch or Loop Currents

Loop_Currents: o

Branch_Currents: 1
i1: 1,2
-

Component VoltageSource Value 15 Component Label V2

Fig 6 Online Tool for drawing of circuits

4. Mentored Problem Solving

As introduced earlier, Internet-based educational systems can be broadly classified in two categories: e-learning systems, and e-tutoring systems. In an e-learning system, a student is given access to learning material in various formats, from text files, to interactive simulation, to animation, and recorded lectures on the subject matter. An e-tutoring system is expected to be a virtual tutor. In a typical tutorial session, a human tutor would verify a student's work at every step of the process, and suggest hints if the student cannot solve a problem or makes a mistake. The IITS attempts to emulate this feature in its "Mentored Problem Solving" environment.

The IITS guides the student to solve a problem in a step-by-step manner following the 'steps' database of the domain knowledge. At the conclusion of each 'step', student's answer is compared with the correct answer. For each step, hints and explanatory figures are available for the student may review. The student can even choose to view additional worked-out examples that elucidate a particular step.

Consider, for example, a student is attempting to solve a circuit using mesh analysis method. There are typically three steps involved: Step 1: write the mesh equations, Step 2: solve the mesh equations, and Step 3: compute various circuit properties, such as, power absorbed by a resistor, voltage across a circuit element, power supplied by the source, etc. For every exercise problem, the instructor breaks up the solution method in a number of steps, and saves the information in the database along with the correct answer for each step. The instructor also saves information about prerequisites for each step. As the student works on an exercise problem, he or she will be prompted with these steps one at a time, and the IITS system would expect an answer from the student. Once the student's answer matches with the answer stored in the database, the student will be prompted to proceed to the next step. If the student cannot find the right answer, feedback will be provided for prerequisite material. Once the student successfully solves the problem, a new problem will be assigned as an exercise.

As the student works 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, and makes the information available to the instructor.

The system uses the information in the student model, and develops a custom exercise for the student. The instructor prepares exercise problems at various levels of difficulty, such as, easy, medium, difficult, very difficult, etc. The student model contains information about the student's level of understanding of the domain. If a student fails to solve a problem, the IITS system will prompt a new easier problem for the student to attempt. If the student successfully completes a few problems at a particular level of difficulty, new problems at a higher level of difficulty will be prompted. 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.

5. Conclusions

The developed course material is fundamentally a first step toward a fully web-based interactive tutoring system. Some progress has been made on various components of the course material, and much more to be completed in future. The best part of the proposed tutoring system is that it is an open-ended system that can be adapted by any instructor using his/her own course material. The instructor has the option to add any number of chapters, sections, and steps as required. The instructor may use any standard file format for uploading the course documents. The shell could be populated at one time before it is made accessible to students, or could be populated slowly over time, for example, over a semester, as the instructor teaches and develops the course.

Acknowledgement:

This research is supported by National Science Foundation's Science, Technology, Engineering, and Mathematics Talent Expansion Program under grant #0230425.

The authors would like to extend their sincere thanks to the reviewers for their constructive comments that have helped improve the presentation of the paper.

Bibliography

[1] Blackborad Inc, http://www.Blackboard.com

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

[3] S. Kuyath, "An Interactive Lecture for Web Based ET Classes", ASEE Annual Conference, Boulder, CO, 2004.

[4] R.G. Daniels, M. Crawford, and M. Mangum, "Web Based Interactive EE Lesson Development: A Modular Approach", ASEE Annual Conference, Boulder, CO, 2004.

[5] B. Butz, "IMITS", College of Engineering, Temple University.

[6] H. Kuehl, E.J. Kazlauskas, and L.O. Picus, "Learning and Cost-Effectiveness of a Web-Enabled Linear Circuits Course", Frontiers in Education Conference, Nashville, TN, 2004

[7] S.-J. Hsieh, P. Hsieh, and D. Zhang, "Web-Based Simulations and Intelligent Tutoring System for Programmable Logic Controller" Frontiers in Education Conference, Nashville, TN, 2004

[8] D. Sharer, M. Frisbee, "Don't Just tell Me, Show Me: Presenting a Microelectronics Course Completely on the Web", ASEE Annual Conference, Boulder, CO, 2004.

[9] J. Porter, and J. Morgan, "A Web Based Instrumentation Platform for use in Distance Learning", ASEE Annual Conference, Boulder, CO, 2004.

[10] S. Kolla and E. Rezaei, "Internet Based on/off controller using LabView", ASEE Annual Conference, Boulder, CO, 2004.

[11] H. Zhuang and S. Morgera, "An Undergraduate Laboratory for Instrumentation and Control", ASEE Annual Conference, Boulder, CO, 2004.

[12] A. Azad, A. Otieno, O. Ghrayeb, N. Anand, "Internet Based Experiments for Physical Laboratory Setup", ASEE Annual Conference, Boulder, CO, 2004.

[13] Chaturvedi S., Osman A., Bawab S., Abdel-Salam T.M., and Venkataramana M. "A Web-Based Multimedia Virtual Experiment", Frontiers in Education Conference, Nashville, TN, 2004

[14] J. Brockman., J. Butler., M. McCready, "A Web Based Learning Tool that Introduces Engineering Concepts by Simulating a Chemical Reactor", ASEE Annual Conference, Boulder, CO, 2004.

[15] Haresh Shahani, Saroj Biswas, Musoke Sendaula, and Harish Pallila, “An Intelligent Interactive Tutoring System for Engineering Education”, ASEE Annual Conference and Exposition 2004, Salt Lake City, Utah, June, 2004

[16] Saroj K. Biswas, Musoke Sendaula, Haresh Shahani, Harish Pallila, “Architecture of an Intelligent Interactive Tutoring System” 33rd ASEE/IEEE/IGIP Conference on Enginenering Education, Fribourg, Switzerland, Sept 27 – Oct 1, 2004.