An Intelligent Tutoring System for Circuit Analysis

Brian P. Butz, *Senior Member, IEEE*, Michael Duarte, and Susan M. Miller

***Abstract*—The Interactive Multimedia Intelligent Tutoring System (IMITS) is designed to assist electrical engineering undergraduate students taking their first circuits courses. The IMITS system places the student in a real-life engineering scenario in which the student is a newly hired engineer within the fictional IMITS Corporation and given "real-life" problems to solve, corresponding to course material. The office has file cabinets, bookshelves, a printer, and a personal computer. The personal computer allows the student to receive televideo messages, receive "e-mail," and send "e-mail" reports to senior engineers. A feature of IMITS is that the student decides which actions to take and may validate analyses and designs using a virtual laboratory incorporated with the software. A brief historical perspective of intelligent tutoring systems is presented, followed by an explanation of their architecture. Next, a detailed discription of the intelligent tutoring system IMITS is given. Then the results of usability and effectiveness evaluations of the software are given.**

***Index Terms*—Expert systems, intelligent tutoring systems, interactive computing, multimedia computing.**

I. Introduction

SINCE the invention of the electronic computer two divergent opinions have existed as to how this device should be used. One group saw the computer's principal use as a number manipulator: an extensive, ultrafast, and accurate calculator. Another group envisioned the computer as a symbol manipulator that might be taught to use logic and make decisions in a human fashion. The symbol manipulation group eventually founded the discipline called artificial intelligence (AI). As time passed, AI spawned many subdisciplines, one of which is expert systems. An expert system is a computer system that performs at or near the level of a human expert in a particular field of endeavor [1].

One of the most interesting applications of expert system technology is for education. The goal of researchers has been to develop a computerized tutor that performs at the level of an excellent human tutor. Computer-aided instruction (CAI) systems, which first came on the scene in the early 1960s, scheduled resources, managed teaching aids, and graded tests. However, the predominant application quickly became using the computer to interact directly with the student rather than have it act as an assistant to the human teacher [2], [3]. Computer-aided instruction systems evolved into intelligent computer-aided instruction systems and then intelligent tutoring systems (ITSs) when principles of artificial intelligence were applied to them. This application occurred in the 1970s and 1980s. Although the development of expert systems for training has been moderately successful and other kinds of expert systems, such as income tax preparation assistants, have been extremely successful commercially, ITS development did not advance significantly for several years after these initial efforts. Within the last decade, prospects for ITS have shown promise. Advancements in computer hardware and software and the appearance of interactive multimedia development tools have created a software environment that makes real intelligent tutoring systems a possibility [4]–[6].

Manuscript received July 19, 2004; revised December 30, 2005. This work was supported in part by the Division of Undergraduate Education, National Science Foundation, under Grants DUE 9752417 and DUE 9952291.

B. P. Butz and M. Duarte are with the Intelligent Systems Application Laboratory, College of Engineering, Temple University, Philadelphia, PA 19122 USA.

S. M. Miller is with the College of Education, Temple University, Philadelphia, PA 19122 USA.

Digital Object Identifier 10.1109/TE.2006.872407

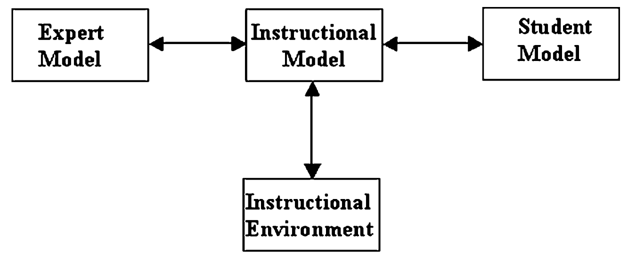

Fig. 1. Traditional intelligent tutoring system architecture.

II. Intelligent Tutoring System Architecture

The power, speed, and relatively low cost of modern technology make computer hardware and its interconnection less and less an impediment to the development of effective intelligent tutoring systems. The largest obstacle is one's ability to design effective interactive learning environments [7]. Intelligent tutoring systems provide their own learning environment and place the student within it. A traditional ITS architecture is shown in Fig. 1. It is divided into four modules: the expert model, the student model, the instructional model, and the learning environment [8].

Most intelligent tutoring systems have their *instructional model* represented in the form of procedural rules [9]. These rules might trigger on certain student errors or on recognized situations encountered in the tutorial session. Other factors that may bring the instructional model into play include the history of the tutorial session (e.g., the student being repeatedly tutored on the same material), significant student actions, the type of knowledge being imparted, and comparisons between the student's and the tutor's knowledge.

The *expert model* is used to ascertain what the student knows, does not know, and knows incorrectly. In communication with the student and instructional models, the expert model often attempts to determine what the student is doing incorrectly. Many expert models also try to identify *why* a student is doing a procedure incorrectly. The determination of why a student does something assumes that the student has learned a procedure for doing something but has learned it incorrectly. Some expert models attempt to identify the student's faulty procedure, sometimes called a bug [10], and to correct it. The rationale for bug identification and correction is so that a student not only will learn how to do a procedure correctly but also will learn how to correct faulty perceptions.

The *student model* should be constructed so that the expert model can use it to determine what the student knows. The expert model needs enough information from the student model so that it knows the student's level of competence, past learning behavior, and learning style(s). The student model is constantly modified as 1) the student proceeds through the software environment and 2) the student takes actions while in the environment.

The *software environment* is the "atmosphere" in which the software developer places the student. The environment needs to be conducive to learning and appropriate to the student's preferred learning style and level of competency. One consideration is that the learning environment should fall within the objectivism–constructivism continuum. An objectivist educator believes in one correct reality and presents material in this manner [11]. Consequently, the course is tightly structured so that often the student is told what to do and when to do it. In 1949, Tyler [12] developed a model based on the objectivist paradigm. Tyler's approach postulates that four steps are necessary for effective instruction: 1) identify the objectives of the instruction; 2) select the useful learning experiences; 3) organize the learning experiences in the best possible manner; and 4) evaluate learning. Constructivism is on the other end of the continuum. Constructivists believe that knowledge does not exist independent of the learner. Thus, knowledge is constructed by the individual. The concept is that an individual observes the world and interprets it. This interpretation, affected by sensory experiences and social interaction, is dynamic and results in meaning to the individual. Consequently, a reasonable software environment might place the user in a situation that is somewhere between no user control (objectivist) and complete user control (constructivist).

III. Interactive Multimedia Intelligent Tutoring System

The Interactive Multimedia Intelligent Tutoring System (IMITS) focuses on the instruction of undergraduate electrical engineering students. The material covered is that typically presented in two courses on introductory circuits, covering dc, ac, and transient analysis. The tutor monitors the student's activities in learning and understanding key material, analyzing any difficulties that the student may be having and tutoring the student when necessary. The tutoring consists of identifying a student's weaknesses in one or more of the learning objectives associated with a particular program of material. (A learning objective is a basic concept that forms an essential body of knowledge in the area being studied. For example, within electrical engineering, the understanding of Kirchoff's voltage law (KVL) and Kirchoff's current law (KCL) is essential to understanding the foundation of circuit theory. Consequently, both KVL and KCL are designated learning objectives within IMITS.) Depending on the nature of the weakness, IMITS then modifies the instructional sequence and/or the amount of detail presented. These modifications are made so that the student may understand any logical flaws, correct them, and clarify any misunderstanding about the material.

Fig. 2. IMITS office.

The IMITS system, developed using Macromedia's Authorware, places the student in a real-life engineering scenario in which the student is a newly hired engineer placed in an office (Fig. 2) of the fictional IMITS Corporation and given "real-life" problems to solve, corresponding to course material. The office has file cabinets, bookshelves, a printer, and a personal computer. The personal computer allows the student to receive televideo messages, receive "e-mail," and send e-mail reports to senior engineers. The first time the student enters the software package, he or she is welcomed to the corporation via a video. Next, the function of the office is explained visually using a voiceover and an interactive demonstration. Then the student receives specific assignments that entail either analyzing designs created by the senior engineers or designing simple systems. The assignments are made via simulated e-mail, televideo messages, or videos. The student has complete flexibility in how to respond by choosing books to consult, by selecting files to read, or by prototyping designs in the virtual laboratory. The files contain basic definitions and specifications about different IMITS Corporation products. Books from the bookshelf contain chapters, which are interactive multimedia tutorial lessons about material that the student should understand to complete the assignment. The students may choose which topic or topics to investigate at any time.

The student also has access to a virtual laboratory (Fig. 3) in which he or she may construct various circuits on a breadboard. The virtual laboratory might be used to test out problem solutions or candidate designs. It might also be used as a prelaboratory exercise that allows students to familiarize themselves with equipment functionality before entering an actual laboratory.

The assignment is at the center of the learning paradigm within the IMITS software package. The assignments may be completed in any order, although preferably, a student should complete them in a linear fashion. As a student attempts to complete an assignment, the expert system watches the progress that a student makes. At prescribed intervals, the student's performance on questions encountered is sent to the expert system. The expert system takes this information, integrates it

Fig. 3. Virtual laboratory.

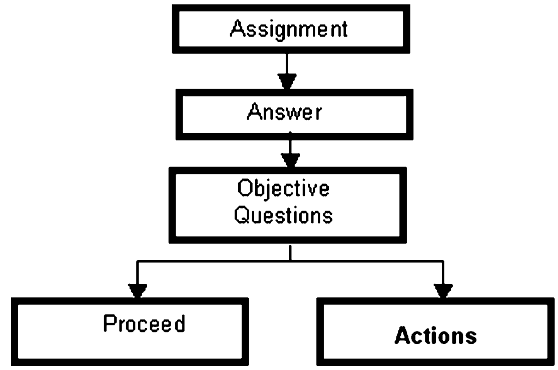

Fig. 4. Assignment scheme.

into the existing model that represents the student's knowledge, and decides what to do next. Decisions include: do nothing, provide additional tutoring on one or more learning objectives, send the student to other interactive material, etc.

A. IMITS Framework

The assignment is the basis of learning within the IMITS software package. More than just the correct answer for the assignment is required of the student. Even if the student completes the assignment correctly, using none of the interactive and questioning features available, specific questions await the student at the end of the assignment. These "objective questions" are based on the learning objectives of the assignment and verify that the student knows how to do the assignment. Thus, the student is evaluated based on his/her knowledge of the tutor's learning objectives, not the correct completion of an assignment.

Various learning objectives are incorporated into every assignment. No assignment contains every learning objective, but several learning objectives occur in multiple assignments. One should consider the assignment scheme shown in Fig. 4. This scheme is used in every assignment in the tutoring system. Fig. 4 shows that the student provides an answer for the assignment. The answer could be correct or incorrect. If incorrect, the student will be helped and will get other chances to complete the assignment correctly (more will be said about this step later). Once the assignment has been completed, the student is given a set of objective questions to answer. These questions focus on knowledge used to complete the steps in the solution of the assignment. The steps concentrate on the combination of learning objectives that need to be mastered to complete the assignment. The objective questions allow the tutor to determine whether the student understands the prerequisite material or stumbled on the correct answer by the fortuitous combination of faulty steps. In addition, if the student did not successfully complete the assignment, the objective questions allow the tutor to determine better what learning objective(s) caused the problem.

Fig. 5 shows in more detail how the student progresses through the assignment to the objective questions. When the virtual team leader e-mails the student's assignment, a hint button is provided. Pressing the hint button allows the student to communicate with the team leader "face-to-face." The team leader, in a video clip, suggests that the student read certain material in a specific book on the bookshelf for additional information. The student can access the hint at any time or ignore it. The student must return to the office to take advantage of the facilities there and complete the assignment. Eventually the student completes the assignment by sending the team leader a virtual e-mail. If the answer is correct, the student is given credit for scoring three points (the maximum) on one question on each of the learning objectives inherent in the assignment. If the answer is incorrect, the student is sent back to the supervisor. The supervisor, through a video, tells the student that the answer is incorrect and then, making use of interactive multimedia, helps the student get started on the solution. This step requires explaining some material and asking basic questions that will start the student on the correct analysis path while maintaining appropriate learning objectives. When the supervisor has finished helping the student, the student is sent back to the office to finish the assignment. The student's performance record (called the student model) is then updated. The number of questions asked for each learning objective and the total number of points earned for these questions are noted.

The student again e-mails the supervisor a solution. If correct, the student receives a credit of two points for answering one question in each of the learning objectives. If the solution is incorrect, the student is again sent to the supervisor. The supervisor helps the student complete a little more of the assignment and employs an interactive multimedia presentation, then asks the student some questions on the material presented. After the coaching given by the supervisor, the student returns to the office, and the student's performance record is updated. For the third and last time, the student is given an opportunity to complete the assignment. If the student's solution this final time is correct, one more question is recorded as being answered for each learning objective. One point is scored for each learning objective for answering this question correctly. If the answer is incorrect, the student is sent to the Skills Advancement Director, a staff member other than the supervisor, who through a video and an interactive presentation shows the student how to complete the assignment.

At this point the student has finished the assignment, either successfully or unsuccessfully. As shown in Fig. 6, the student is next presented a sequence of objective questions that go over

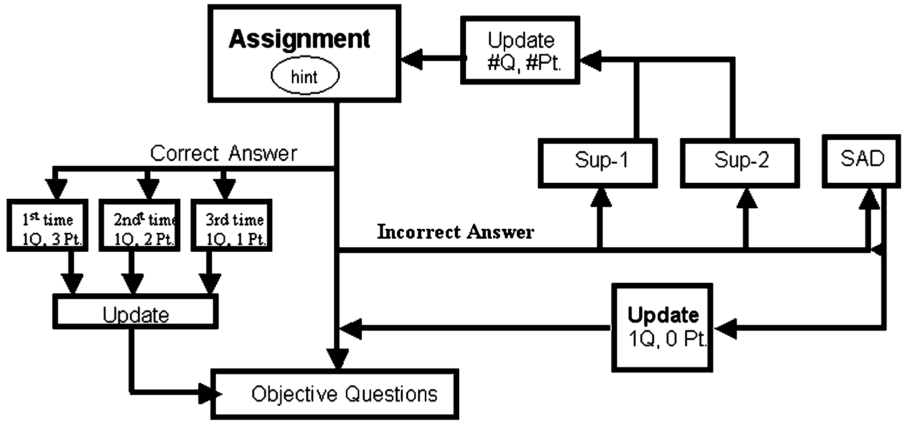

Fig. 5. Detailed look at each assignment.

TABLE I
DATA PASSED BETWEEN INTERACTIVE MULTIMEDIA AND EXPERT SYSTEM

Authorware	**Expert System**
Information passed	*Student model*
Learning objective k	**Learning objective k**
-number of questions -score -possible score	-updates number of questions -updates score -updates possible score -moves past delta to historical delta -moves recent delta to past delta -calculates recent delta (pct.) -cumulative questions -cumulative score -cumulative possible score
Modules visited	
-Bookshelf modules visited	*Bookshelf modules* -list of modules visited
- Tutoring modules visited	*Tutoring modules* -list of modules visited

basic aspects of the assignment. Upon completion of the objective questions, the student's performance evaluation is updated and sent to the expert system.

The expert system decides if the student's performance is acceptable or unacceptable. Each learning objective encountered by the student so far is examined. If a student's score for a learning objective is 60% or higher, the student's performance is acceptable for that learning objective. If a student's performance on *every* learning objective encountered thus far is acceptable, the student is sent on to the next assignment. If a student scores less than 60% on one or more learning objectives, the student needs additional work on those learning objectives. Next, the expert system examines the student model and decides what further actions it can take to assist the student.

B. Information Exchange

As shown in Table I, information is passed from the Authorware program to the student model at predefined intervals. Time tags are added at the beginning and the end of the student model each time the student enters and exits IMITS. The first item of the student file will be the date and time in; the last item of the student file will be the date and time out. As the student proceeds through the various Authorware modules, files are dynamically created. These files track the various submodules that the student visits, record and store the student's response to each interaction, and note the learning objective attached to each interaction. The expert system then mathematically combines information if a learning objective is encountered more than once as a student proceeds through the virtual environment. The IMITS expert system receives information from the Authorware core program at certain time intervals. Information is transmitted when the following occur:

- a bookshelf module is completed or exited properly;
- the end of the objective questions of an assignment is reached;
- the end of a tutorial is reached.

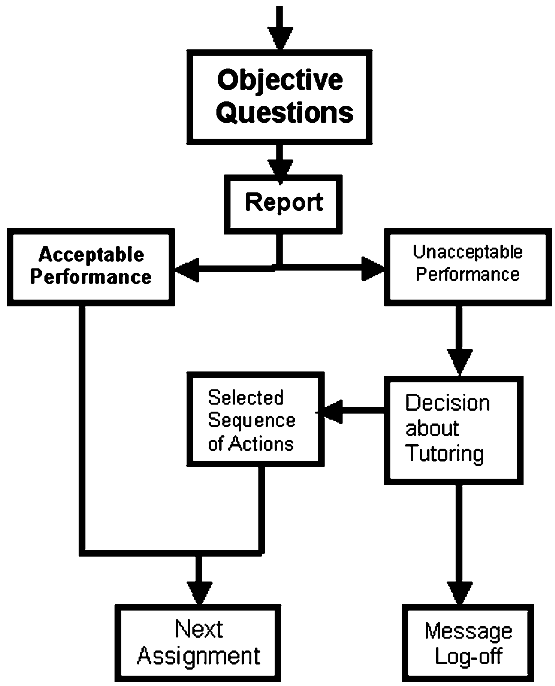

Fig. 6. Action based on student's performance.

A bookshelf module is completed if the student completes an entire chapter or subchapter of a book. The module is exited properly if it is completed or if the student clicks on the office button (returns to office) while in a chapter or subchapter of a book.

C. Expert System

IMITS' expert system replicates the actions of a very good instructor or tutor. Its role is to "watch" the student's performance and to determine the student's knowledge level. It also determines the student's proficiency in essential areas of the material covered. If a student is having difficulty in one or more essential aspects of the course material, the expert system determines what action(s) to take.

Often students are required to answer questions or provide solutions to problems. Each student response requires the understanding of one, and only one, learning objective. Making the learning objectives independent of one another allows the system to measure more accurately the student's mastery of each learning objective. In addition, framing questions one learning objective at a time shapes the pedagogy so that the student is shown a logical systematic method to solve a problem.

As the student interacts with the material, a *student file* is generated and is periodically updated. The student file contains information about how a student is scoring on each learning objective and what modules within the IMITS software environment the student has visited. Specifically, the communication between the expert system and the Authorware core program is shown in Table I.

When the expert system receives these data, it operates on them by scanning every learning objective found in the student model. It checks the cumulative score achieved for each learning objective (LO). It notes each learning objective whose cumulative score is less than 60%. Each learning objective is ranked by its level of sophistication. For example, suppose a student is having trouble with Kirchoff's voltage law, mesh current analysis, and Ohm's law. Kirchoff's voltage law is considered the most fundamental concept of the three. Next in rank is mesh current analysis, followed by Ohm's law. The expert system would first deal with Kirchoff's voltage law, then with mesh current analysis, then Ohm's law. What follows is a description of how the expert system handles each learning objective whose cumulative score is less than acceptable.

The expert system first determines whether a student has been through the objective questions that appear at the end of the assignment. If a student has completed the objective questions, the expert system sends the student to the bookshelf or decides whether to give the student tutoring. If a student has not completed the objective questions associated with the assignment, the expert system determines whether the student has received any tutoring to date. If the student has not yet received tutoring on that LO, the expert system sends a command to Authorware to give the student tutoring on the LO. After tutoring, the expert system reevaluates the student's performance on that LO. If the student already has received tutoring on a deficient learning objective, the expert system looks at how well the student performed on that LO when the last set of information was transferred from Authorware to the expert system. If the last report (called the recent delta) on that LO showed a student score equal to or greater than 60%, the expert system takes no action on that LO. In this case the value of the recent delta convinces the expert system that the student is showing significant improvement on that LO.

If the student's performance on the LO is unacceptable and if the student has received tutoring but the student's recent delta on that LO is less than 60%, the expert system has some decisions to make. Has the student been tutored less than three times? If the answer is yes, the student receives tutoring again. If the student has already been tutored three times, the expert system ascertains whether the student has been to the bookshelf that contains material relevant to that LO. If the student has been to the bookshelf but has not been to the objective questions associated with the assignment, the student is sent to the assignment so that the student may complete it. If the student has completed the assignment (been to the objective questions), has been to the bookshelf, and has been tutored by the expert system three times, the student has run out of options. The expert system directs the virtual supervisor to dismiss the student from the IMITS Corporation. When a student is dismissed, he/she must see the course instructor who will have received a report from the program concerning the student's performance. After the instructor assists the student and is confident that the student understands the learning objective(s), the student is given a code and is able to use the software (return to the company) again. The student's performance indicators are reset to satisfactory in those areas that caused the "termination."

D. Laboratory

The virtual laboratory, which has been described in detail elsewhere [13], [14], consists of a breadboard, components, and instruments. Fig. 7 shows the workspace given to the student. The virtual laboratory users may build and test circuitry with

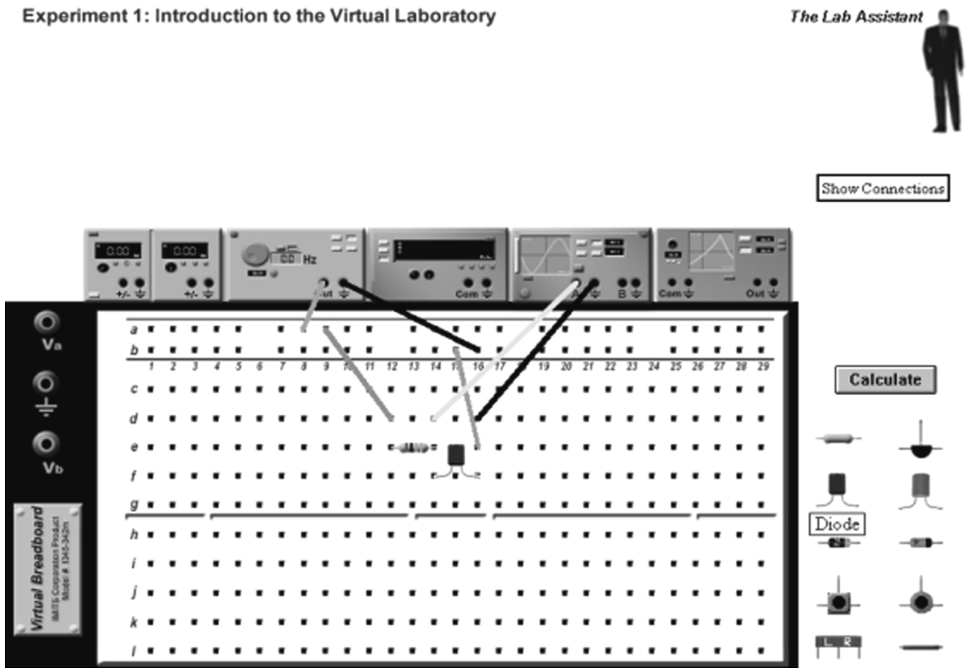

Fig. 7. The laboratory interface.

the following components: resistors, capacitors, inductors, variable resistors, variable capacitors, diodes, zener diodes, transistors, and jumper wires. In addition, the laboratory contains two dc power supplies, a function generator, a multimeter, a two-channel oscilloscope, and a spectrum analyzer with a sweep generator.

IV. Evaluation of IMITS

The evaluation [15] of the IMITS software environment occurred in two phases: a usability assessment and an effectiveness evaluation. Usability data provide information on a program's functional effectiveness, efficiency, and ease of learning and use, motivational influence, and quality assurance. Effectiveness evaluation focuses on a program's impact on student learning.

A. Usability Evaluation

1) Sample and Instrumentation: Usability data were obtained from students in a single group design and from students who were part of the effectiveness study. For the usability evaluation, participating institutions included Duke, Howard, New Mexico State, and Temple Universities, and Montgomery County (PA) Community College; Temple and Rowan Universities were involved in the effectiveness study. In all, 114 students completed the *IMITS Usability Questionnaire* that included 22 items using a five-point Likert-type scale (where $1 =$ strongly disagree and $5 =$ strongly agree) and open-ended items. Usability information was also obtained from user log files.

2) Usability Results: Students rated IMITS as a useful learning tool regarding acquisition of engineering concepts ($\text{mean} = 3.58$) and as an expert source of information (mean $= 3.36$). About half of the students thought the more they used the software, the more they learned. Although students are not as likely to agree they enjoy a learning product as much as they do a leisure one, the evaluator directly asked them this question ($\text{mean} = 3.11$). Most students found IMITS easy to use ($\text{mean} = 4.06$) and intuitive ($\text{mean} = 3.68$). They gave the same high marks to using the different office components ($\text{mean} = 3.77$) and to using the virtual lab ($\text{mean} = 3.15$). They reported limited navigation within sections ($\text{mean} = 3.12$).

Students reported that the various components of the software program were helpful; they rated the virtual bookshelf as the most helpful feature (58%). Eighty percent highly endorsed the clarity of the book materials. Although participating students were likely to be technologically sophisticated, they rated several features of the virtual environment as realistic: student's role as a junior engineer ($\text{mean} = 3.07$), the company office ($\text{mean} = 3.05$), and virtual lab ($\text{mean} = 3.04$). Students rated IMITS as a high-quality multimedia product including audio quality ($\text{mean} = 3.99$) and graphics quality ($\text{mean} = 3.83$).

Analysis of students' log files indicated that students covered a substantial number of learning objectives ($\text{median} = 11$). The most frequently encountered objectives related to Ohm's law, Kirchoff's current law, and Kirchoff's voltage law. The least frequently encountered were remedial objectives related to complex algebra and graph reading. This pattern is consistent with content expected in the first semester of a beginning circuits course.

B. Effectiveness Evaluation

Two major questions guided the evaluation of the software's effectiveness on learning: 1) Did students who used IMITS (experimental condition) learn more (i.e., score higher on achievement measures) than their counterparts in the control condition? and 2) Did performance on the IMITS embedded learning objective questions improve the more the software was used?

TABLE II
STATISTICALLY SIGNIFICANT DIFFERENCES BETWEEN EXPERIMENTAL AND CONTROL GROUPS ON LEARNING MEASURE

	Experimental (IMITS	Control (No IMITS)	
	Mean SD	Mean SD	F and Probability
Cohort 1			
Course Lab #4	9.17 (2.8)	3.33 (4.8)	F(1,37) = 22.45 p. = .000
IMITS Problem #5	94.58 (5.8)	87.80 (4.7)	F(1,37) = 16.16 p. = .000
Cohort 2			
Course Lab #4	79.9 (36)	91.8 (23)	F(1,34) = 5.17 p. = .029
Course Exam #3	78.79 (20)	65.94 (18)	F(1,34) = 4.90 p. = .034
Cohort 3			
IMITS Problem #3	7.96 (2.0)	3.9 (3.9)	F(1,47) = 21.74 p. = .000

1) Sample and Design: Students enrolled in four introductory circuits courses at two institutions participated in a quasi-experimental study to examine the impact of IMITS on learning ($N = 175$). Each course had two sections: students in one section composed the control group and students in the other section received a curriculum in which the instructor integrated IMITS (experimental $n = 90$, control $n = 81$). The intervention was randomly assigned to matching sections. Each corresponding experimental and control group was referred to as a cohort.

2) Data Analyses and Results: Between subjects analyses using analyses of variance (with grade point average as a covariant for Cohorts 2 and 4) were used to compare classroom achievement measures. As instructors used different assessment measures, analyses were conducted by cohort. For all differences that were statistically significant, students in the experimental group scored higher on performance measures than those in the control group (see Table II).

Combining data from all students' log files, 16 separate ordinary least squares regression analyses were used to capture the direct effect of the number of questions on each objective encountered on gain scores for each learning objective. In addition to the number of questions encountered, grade point average, race, and school were also regressed on the gain score. School was included in the model for only 11 analyses, because many had either no or too few students in all courses that addressed these five objectives.

For 12 learning objectives, the number of questions encountered by a student was directly related to the learning gain score. For nine of these, the number of questions encountered was the only statistically significant predictor variable. For the other three learning objectives, race and/or school were also parts of the predictive model. Details on the regression analyses are available from the authors.

V. CONCLUSION

This paper has described a unique intelligent tutoring system, the Interactive Multimedia Intelligent Tutoring System, which has the following characteristics.

- It is student-centered. It allows the student to decide how he/she wants to approach the assignment. It allows the student to use, in any order, any or all of the resources present in the software environment.
- The pedagogy is one of "guided constructivism." The assignment structure moves the student through an introductory circuits curriculum. The assignments build continuously upon previously learned concepts.
- Each student interaction is associated with one, and only one, learning objective. The expert system can pinpoint an area of student deficiency. The deficiency is remedied by a learn-by-doing approach. Multiple avenues of coaching and tutoring are presented to the student that allow the student to see the concept employed and to understand the concept itself.
- The student has an opportunity to use a virtual laboratory to see if the proposed solution works in actuality.
- The entire IMITS framework provides a structure in which other interactive tutoring systems can be created. The concepts of learning objectives and tutorials are constant across various academic tutoring systems. Specific learning objectives need to be selected for each course tutored, but the scoring and decision-making process is constant across academic systems. The structure of the expert system coding remains constant, as do the elements that compose the student model.

REFERENCES

[1] B. P. Butz, "Expert systems," *Abacus*, vol. 5, no. 1, pp. 30–44, 1987.

[2] A. Barr and E. Feigenbaum, *The Handbook of Artificial Intelligence*. Reading, MA: Addison-Wesley, 1982, vol. 2, ch. 9.
[3] D. Sleeman and J. S. Brown, Eds., *Intelligent Tutoring Systems*. New York: Academic, 1982.
[4] E. Wenger, *Artificial Intelligence and Tutoring Systems*. San Francisco, CA: Morgan Kaufmann, 1987.
[5] M. Polson and J. J. Richardson, Eds., *Foundations of Intelligent Tutoring Systems*. New York: Addison-Wesley, 1988.
[6] K. Van Lehn, "Conceptual and meta learning during coached problem solving," in *Proc. 3rd Int. Conf. Intelligent Tutoring Systems (ITS)*, C. Frasson, G Gauthier, and A. Lesgold, Eds. New York: Springer-Verlag, 1996.
[7] M. O. Hagler and W. M. Marcy, "Computer-aided instruction," in *Wiley Encyclopedia of Electrical and Electronics Engineering*, J. G. Webster, Ed. New York: Wiley, 1999, vol. 3, pp. 648–668.
[8] A. S. Wolff *et al.*, "Using quasiexperimentation to gather design information for intelligent tutoring systems," in *Facilitating the Development and Use of Interactive Learning Environments*, C. P. Bloom and R. B. Loftin, Eds. Mahwah, NJ: Lawrence Erlbaum, 1998, pp. 21–49.
[9] J. W. Rickel, "Intelligent computer-aided instruction: A survey organized around system components," *IEEE Trans. Syst., Man, Cybern.*, vol. 19, no. 1, pp. 40–57, Jan.–Feb. 1989.
[10] J. S. Brown and R. R. Burton, "Diagnostic models for procedural bugs in basic mathematical skills," *Cogn. Sci.*, vol. 2, pp. 155–192, 1978.
[11] C. Vrasidas, "Constructivism vs. objectivism: Implications for interaction, course design and evaluation in distance education," *Int. J. Educ. Telecommun.*, vol. 6, no. 4, pp. 339–362, 2000.
[12] R. W. Tyler, *Basic Principles of Curriculum and Instruction*. Chicago, IL: Univ. of Chicago Press, 1949.
[13] M. Duarte and B. P. Butz, "An intelligent universal virtual laboratory," presented at the 34th Southeastern Symp. System Theory (SST), Huntsville, AL, Mar. 18–19, 2002.
[14] ——, "The virtual laboratory for the disabled," in *Proc. Frontiers Education 2001 Conf.*, 2001, pp. SIC-23–SIC-26.
[15] B. P. Butz and S. M. Miller, "Improving learning of engineering students: demonstration of interactive multimedia intelligent tutoring system," presented at the 2003 AERA Annu. Meeting, Chicago, IL, Apr. 21–23, 2003.

Brian P. Butz (S'69–M'71–SM'94) received the Ph.D. degree in electrical engineering from Drexel University, Philadelphia, PA.

In 1987, he founded the Intelligent Systems Application Laboratory (ISAL) at Temple University, Philadelphia, PA. This Laboratory provided a focal point within Temple University, concentrating on research in intelligent systems. From 1989 to 1996, he was Chairman of the Electrical and Computer Engineering Department, Temple University, where he is currently a Professor of Electrical and Computer Engineering.

Dr. Butz is a member of the American Society for Engineering Education (ASEE) and the Association for the Advancement of Computing in Education. He has been a member of the Executive Committee of the Philadelphia Section of IEEE and was its Chair in 2000. He has received ASEE's Centennial Certificate, IEEE's Third Millennium Medal, and the IEEE Philadelphia Section Award. He received the Christian R. and Mary F. Lindback Award for Distinguished Teaching at Temple University in 1984. In April 1999, he received the Innovative Excellence in Teaching, Learning and Technology Award at the Tenth International Conference on College Teaching and Learning. He received the Distinguished Alumnus Award from the Electrical and Computer Engineering Department, Drexel University, and the Temple University Great Teacher Award in 2000.

Michael Duarte received the B.S.E. and M.S.E. and degrees from the Electrical and Computer Engineering Department, Temple University, Philadelphia, PA.

He is a Software Engineer with the Intelligent Systems Application Center (ISAC), Temple University. Since fall 1997, he has been one of the major contributors to the Interactive Multimedia Intelligent Tutoring System (IMITS) and the Universal Virtual Laboratory (UVL). He has published several papers about the Universal Virtual Laboratory.

Mr. Duarte is a member of the Golden Key National Honor Society and Eta Kappa Nu. He received the Connelly Foundation Scholarship, the Junior Achievement Scholarship, and a Presidential Fellowship. He is listed in *Who's Who Among Colleges and Universities*.

Susan M. Miller received the B.A. degree in psychology from the University of Chicago, Chicago, IL, and the M.S. degree in instructional design and Ph.D. degree in educational psychology from Purdue University, West Lafayette, IN.

She is an Assistant Professor in educational psychology at Temple University, Philadelphia, PA, and Director of a Master's degree program in instructional and learning technology. In her role as Educational Psychologist and Program Evaluator, she has been affiliated with the Intelligent Systems Application Center for several years. Her professional interests include the application of theory and consideration of individual differences in technology-rich environments and evaluation of intelligent agents. She has conducted research in the area of multicultural education and coauthored the "Survey of Cultural Attitudes and Behaviors" designed to assess cultural bias and discrimination in educational settings. She has published in these areas, as well as made numerous regional, national, and international professional presentations.

Dr. Miller is a member of the American Educational Research Association and the American Psychology Association.