Freedom of Choice in an Intelligent Tutoring System*

Brian P. Butz
Electrical and Computer Engineering Department
Temple University
Philadelphia, PA 19122

Abstract

An Interactive Multimedia Intelligent Tutoring System (IMITS) is described. IMITS has been constructed to assist students learn difficult portions of three electrical engineering undergraduate courses. The multimedia tutor possesses intelligence provided by an expert system that watches what a student does and what answers a student provides to tasks and questions. From the analysis of the student's responses, the expert system infers the level of the student's knowledge about the subject area and modifies dynamically the presentation and level of the interactive material.

1. Introduction

Interactive multimedia educational systems are becoming increasingly accepted as another technique in helping a student learn better and more efficiently. A logical next step is to place intelligence within multimedia presentations so that an intelligent tutoring system (ITS) results. The ensuing ITS contains a multimedia presentation, an expert system, and an invisible communication channel between them. The expert system, the intelligence of the Intelligent Tutoring System, "watches" the student, determines what the student knows, does not know and knows incorrectly, and uses this information to modify the presentation of the multimedia material.

Today, more is expected from software than mere functionality. Not only must software achieve its intended objectives but it also must be engaging. In the past intelligent tutoring systems presented students with a series of questions, carefully crafted problems, or directed actions which the students were asked to answer or perform. Once the student completed the tasks, the ITS analyzed the results to determine the student's progress. This paper describes an ITS that places the student in a virtual office, presents the student with a "real life" problem to solve, then allows the student to choose whatever action is deemed appropriate. The problem then becomes: given that a student may do a finite number of things, can the expert system analyze what is going on and what tutoring a student needs on what topics? Because this paper is written as the project is developing and well in advance of its presentation, it will be

---

*Partial support for this work was provided by the National Science Foundation's Division of Undergraduate Education through grant DUE #9752417.

concerned with the planning associated with the implementation of the expert system. The presentation will demonstrate the system.

2. Interactive multimedia

A continual challenge in teaching is meeting the students' academic needs and ensuring that they are active participants in their own education. Over the past decade several studies have been made to detect students' learning styles [1-5]. These studies conclude that students best learn if they are active participants in their education and are exposed, in varying degrees, to each discerned learning style. Visualizing the dynamic behavior of physical systems is an important part of the learning process for engineering students [6]. Interactive multimedia presents an opportunity for using advanced technology to improve and enhance a student's learning environment [7,8]. Students are more attentive to visual lessons, seem to retain more of what they see over what they hear, and are more comfortable with visual learning . While texts are certainly visual, they tend to be static while computer software and computer networks have introduced students to dynamic visual material that may be modified by the user.

Interactive multimedia software can play a key and unique role in the educational process. It may be used as a stand-alone educational module that is intended to enable an interested individual to learn about a particular topic or subject [9]. It may be used as a supplement to classroom presentations [10] and engineering laboratories [11], or it may be used as a dynamic textbook [12]. Recently interactive multimedia has become an effective way of educating disabled students [13-15]. Interactive multimedia brings an additional dimension to on-line learning (OLL) courses. OLL courses, which use the world wide web to present material, animations , movies and other forms of user interaction, are a growing subset of more traditional distance learning programs. Because of technological advancements the number of universities offering distance learning programs has increased from 94 in 1993 to 384 in 1996 [16]. Many of the newer distance learning programs offer at least some of their courses asynchronously and the asynchronous offerings are mostly on-line. The explosion in OLL has lead to a Journal of Asynchronous Learning Networks [17] and numerous examples of OLL using interactive multimedia may be found [5, 9, 18]. Moreover, interactive multimedia software is currently being marketed as a stand-alone asynchronous learning aid. Examples include various members of the Schaum's Outline Series [19-20], an electronics laboratory simulator [11], and a set of introductory calculus modules [21]. Also the National Science Foundation has funded scores of investigators to develop interactive multimedia educational modules or interactive multimedia courseware in the past and will no doubt do so in the future.

Consequently, it is reasonable to assume that interactive multimedia modules are no longer simply interesting curiosities. It is very likely they will become part of the 'usual' educational process in the foreseeable future especially as the authoring software becomes easier to use and becomes more powerful leading to a reduction in the long development time required for each educational module [22].

3. Intelligent tutoring systems

The forbearers to intelligent tutoring systems were computer-aided instruction (CAI) systems which first came on the scene in the early 1960's. Originally these applications included scheduling resources, managing teaching aids, and grading 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 [23]. The first CAI programs were either computerized versions of textbooks (characterized as electronic page turners) or drill and practice monitors that presented a student with problems and compared the student's responses to the pre-scored answers.

Computer-aided instruction systems evolved into intelligent computer-aided instruction (ICAI) systems and, then, intelligent tutoring systems (ITS) when principles of artificial intelligence were applied to them. This occurred in the 1970's and 1980's. Both ICAI and ITS contain explicit knowledge of the subject taught. ITS also attempts to simulate the behavior of an intelligent human tutor in addition to acting as a domain expert. The characteristics of an ITS include the ability to teach a given subject, to detect student errors, to try to figure out where and how the student made an error, to correct flaws in the student's logic and to clear up any misgivings or misunderstandings the student may have about the material [24-25].

The models used today for ITS are based upon early models such as BUGGY, developed by John Seeley Brown (of Xerox PARC) and Richard Burton and Kathy Larkin (researchers at Bolt, Beranek and Newman, Inc.)[26] and others including EXCHECK [27], SOPHIE [28] and GUIDON [29]. Eventually the research that developed ITS resulted in a number of training systems that were used by companies to train employees. One system was STEAMER [30] which was designed to teach naval officers about the problems of running a steam propulsion plant and DELTA [31] which assisted General Electric's mechanics in diagnosing and repairing problems in diesel-electric locomotives.

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 implementation did not advance significantly since the 1980's. It is only recently that prospects for ITS have shown promise. Advancements in software over the last decade along with interactive multimedia development tools have created a software environment that make real Intelligent Tutoring Systems a possibility [32 - 34].

4. The design of the Interactive Multimedia Intelligent Tutoring System (IMITS)

During the past decade the percentage of students who purchase the required text for a university course has declined. Moreover, in many instances those students who do buy the required text do not read it sequentially but use it as a reference manual to provide answers to specific questions. At Temple University this has resulted in a significant increase in student requests for tutors. Because of the increased demand it made sense to investigate the possibility of developing a software-based tutoring system that 1) presented problematical material to students in an interactive setting, 2) contained sufficient expertise in the material presented to be able to analyze a student's actions and to change the presentation to suit the student's level of understanding , 3) engaged the student by placing the student within an engineering office and requiring the student to proceed in a professional manner and, 4) underlined the importance of the written word in understanding the theory underlying the various assignments. The resulting software package that is being developed is called the Interactive Multimedia Intelligent Tutoring System (IMITS).

The philosophy behind the intelligent tutor is that is should function much like an instructor with whom the student visits and requests help solving electrical engineering problems. The instructor gives the student a problem in the area in which the student is having difficulty and watches the student's performance. The student, the instructor or both may ask questions as the problem solving proceeds. The instructor may see that the student has difficulty with a particular concept and put the problem on "hold" while asking the student specific questions about the concept until the student shows an understanding of it. If the student gets stuck on the problem, the instructor might ask the student questions that lead the student back on track and allow the student to resume the problem solution.

An overview of the tutoring system, IMITS, is shown in Figure 1. The IMITS software package is actually a collection of rather large applications that have been programmed individually to perform specific functions and to communicate, as needed, with the other application packages. The student, using a personal computer, enters the software package and begins an interactive module that has been developed using Macromedia's Authorware 4.0 [35] development software. As the student progresses through the module every interaction the student makes (every selection, change of selection, computation, etc.) is monitored and recorded by Authorware in the results file. All of this information is available to another software package, the expert system. Within IMITS the expert system[**] analyzes the results at appropriate intervals during the interactive session to determine how well the student understands and applies the material presented in the module. If the expert system detects a problem in student learning it may reconfigure the session by asking the student simpler or more detailed questions or it may decide to let the module proceed so that it may obtain more information about the student's thought process.

When the expert system has enough information to determine any procedural or logical flaw exhibited by the student the interactive module is modified. The student continues to interact and is oblivious to the presence of the expert system (the student-expert system interaction appears to the student to be part of the learning module). The learning module is reformulated until the expert system detects a satisfactory performance in the student's understanding of the material. If a satisfactory level of understanding is not obtained the expert system generates a file for the student and another file for the instructor. The student's file

---

[**]In general, an expert system is a computer system that performs at or near the level of a human expert in a particular field of endeavor [36].

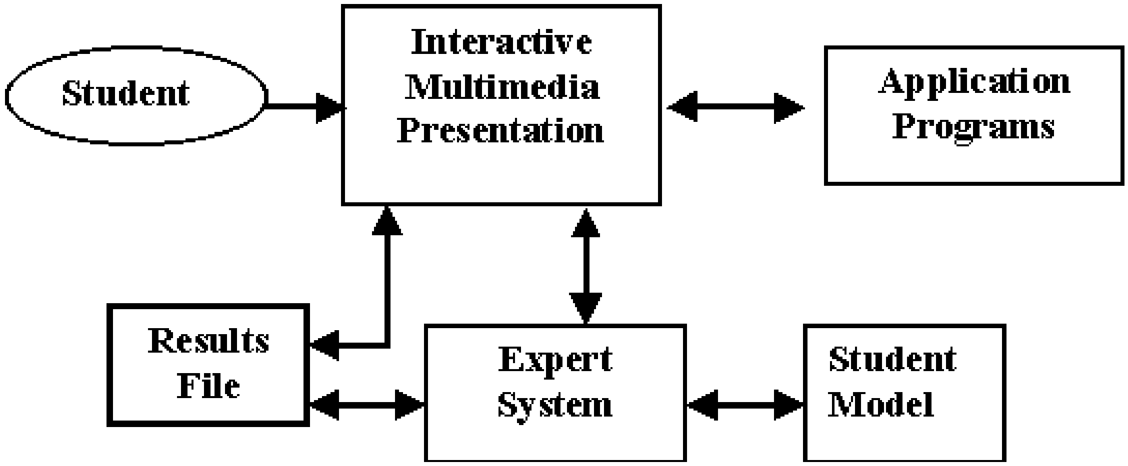

Figure 1 Overview of Interactive Multimedia Intelligent Tutoring System

contains suggestions for improvement and directs the student to the instructor. The instructor's file contains a detailed analysis of the student's performance for the instructor's use. The student model is a creation of the expert system and represents the student's level of knowledge about the material being tutored. The student model contains a representation of the student's understanding of the different learning objectives presented during the tutorial. For example, the expert system may determine that a student understands what a time constant (one of the learning objectives) is and how to use it and will place this assessment in the student model.

Macromedia's Authorware was chosen as the multimedia development tool because it has a relatively short learning curve for its basic use, it is a cross platform tool so it will run on both PC's and Apple computers, and because the resulting modules can be packaged as executables so that a user need not have Authorware to run the modules. Authorware is able to exchange information dynamically with other application programs and is noted for its ability to judge, track and record. The expert system development software (called a shell) is CLIPS [37] a software package developed by the National Aeronautics Space Administration.

5. The expert system

The IMITS project concentrates on parts of three electrical engineering undergraduate courses. Two of the courses, Electrical Engineering Science I and II, are typical sophomore level circuits courses taken by electrical engineering undergraduates. The material covered consists of DC, AC, and transient analysis as well as frequency response and Laplace transforms. The third course is a junior level classical control course taken be electrical engineering undergraduates. Within the interactive multimedia presentations there are various learning meta modules[***] that deal with discrete areas of instruction. Each meta module has a paradigm associated with it. For example, there is an AC Laboratory meta module which assigns the student tasks that are performed within a virtual laboratory . Similarly, the transient analysis meta deals with the transient analysis of circuits and places the student in a virtual office of a fictional corporation.. It is this transient analysis meta module and the role of the expert system within it that will now be considered. The transient analysis meta module is used by first semester second year undergraduates students in the first course in electrical circuits.

Upon entering the transient analysis meta module the student is placed in a virtual office (Figure 2). The session begins with a video that shows a situation. Then the office appears with the computer ringing. The student has been told that this sound represents an incoming teleconference call. By clicking on the computer screen, the student encounters several icons including a teleconference one. Double clicking on that results in a "teleconference" hookup (actually an avi movie) with the Vice-President for engineering who describes the student's assignment. The student then has access to 1)the file cabinets, which contain brief technical notes, 2) the books on the bookshelves that contain interactive exercises that help a student refresh his or her recollection of basic principles essential to the solution of the problem, 3) the computer which allows the student to teleconference, email, make notes and save them, and launch other applications such as PSPICE (a circuit analysis program), and 4) the printer which allows the student to print out the technical notes and specifications found in the file cabinet.

In order to plan an expert system for the office paradigm several postulates were made. The material in the files would be brief, containing the specifications necessary to do the problem. There would be no interactivity with the files other than reading and printing them at will. The books contain highly interactive materials broken down into modules such as *time constants, initial conditions*, and *first order circuits*. The information in these modules is carefully crafted so that students answer questions that should allow evaluation of their knowledge of the area covered by the module as well as the immediate prerequisites to the material being reviewed. Each answer, click, typed response is recorded by the authoring software and placed into files which are reviewed by the expert system at programmed intervals. Feedback is given to the student and the student is shown the correct answer to questions after at least two attempts are made by the student. Since the student has accessed the books seeking help, help will always be given to a student that needs it. The student may also access the computer to get each assignment by "teleconferencing". The email icon is used to submit a completed report to the Vice President of Engineering. Once the report is received, a video clip is sent back to the student showing how the design was used and what was the result of its use.

The expert system interacts with both the *results file* and the *student model* to determine what action it should take to assist the student. The results file consists of a map describing how

---

[***] A meta module is a module that oversees and organizes several modules that deal with a specific learning objective.

Figure 2 The Virtual Office of the IMITS Corporation

the student has steered himself or herself through the system as well as every response the student makes. Also recorded are the times spent on each question and each module. The student model is a dynamic representation of what knowledge the student possesses in the area of interest. It is modified by the expert system as it evaluates a student's progress by scanning the results file. Consequently, it can determine if a student's comprehension, as measured by responses, is increasing, decreasing, or staying the same. It also notes material that the student handles easily and what material poses problems.

After modifying the student model, the expert system uses it to decide what action to take. Actions include: doing nothing, calling up and implementing remedial modules, deciding to skip material in the chapters of the book that a student knows well, bringing in animations to illustrate more fully a point or to clear up detected confusion or misunderstandings, launching an application program to demonstrate a point and then to have the student use this application program to reinforce learning of this point. All of these actions need to appear seamless to the student. If a remedial module is presented, the student is not told that it is a remedial module -- it just appears.

The expert system shell used in this project is CLIPS 6.04 (C Language Integrated Production System). CLIPS is a forward chaining[****] production system[*****] that was written by NASA. Its inference engine contains truth maintenance, dynamic rule addition and customizable conflict resolution strategies. It has its own object oriented language called COOL which is directly integrated with the inference engine. CLIPS is well documented [38], it is well supported, it is inexpensive and runs on most platforms. The forward chaining technique used by CLIPS will ensure that it concentrates on what it perceives that the student knows or does not know and not on the fallacy or bug that has brought the student into error. The rule base[******] will be constructed and modularized so that the expert system does not cause delays leading to user frustration. The expert system knowledge base[*******] and rule base is developed as a result of an iterative process.

During the Fall 1998 semester students in the introductory course used the interactive multimedia transient meta module without the expert system. Each student kept a diary of his/her thoughts when using the module and answered a list of questions about the module when the student had completed it. The questions and the diary were used to determine what was good and what was bad about the module, what was clear and what was confusing and what was good interactivity and what interactivity could be improved. The students were also asked for what they would like to see in the meta module that was not there. Simultaneously, the tracking ability of Authorware was used to record what a student did and how long it took to accomplish each task the student decided to do. These results representing what students did in what order helped form a basis for the rule base. The knowledge base was formed using the expertise of the author and his experience with tutoring students in and out of the classroom.

[****] Forward chaining is a control strategy that regulates the order in which inferences are drawn by asserting all the rules whose "if" clauses are satisfied. Given these assertions, the strategy determines what other rules may now be true. This procedure continues until the program reaches a goal or runs out of satisfied rules.

[*****] A production system is a rule-based system containing "if-then" statements as its database.

[******] A rule-based system is one that acts when at least one rule is satisfied. a rule is a statement consisting of two parts, an *antecedent* and a *consequent*. The antecedent consists of one or more "if" clauses, and establishes conditions that must be met if the consequent part of the rule is to be activated. The consequent is composed of the actions or conclusions that result (e.g. if it is raining and it is cool then wear a raincoat).

[*******] A knowledge base is the part of an expert system that contains the facts and heuristics about a specific domain.

6. Conclusions

This paper described the planning that went into the development of the expert system that oversees a student's progress in a loosely structured meta module of an intelligent tutoring system. The module described was the transient response meta module. Its virtual office paradigm not only presents the students problems appropriate to an introductory course but also gets them familiar with typical engineering tasks such as looking up specifications, reading material on the task at hand as required, bringing into play application programs, and delivering their results in a report form.

The expert system watches what a student does and in what order he or she does it. With the knowledge ascertained from the student's results file and the dynamic student model, the expert system makes certain tactical decisions about what should be done to assist the student. The expert systems continues to monitor student progress to determine if its actions have been effective and if the student continues to progress.

7. Acknowledgements

The meta modules described in this paper and presented during the demonstration were developed by a team of extremely dedicated and creative Temple University students: Michael Duarte, Zhongxin Li, and Jereme Lovekin of the Electrical and Computer Engineering Department, Matthew Bartels of the Broadcasting, Mass Media and Telecommunications Department, and Edward Gaittins of the Esther Boyer College of Music.

Bibliography

1. Felder, R., Matters of Style. *ASEE Prism*, 6 (4), 18-23 (1996).
2. Lumsdaine, M., and Lumsdaine, E. , Thinking Preferences of Engineering Students: Implications for Curriculum Restructuring. *Journal of Engineering Education*, 84 (2), 193-204 (1987).
3. Stice, J.E., Using Kolb's Learning Cycle to Improve Student Learning. *Engineering Education*, 77, 291-296 (1987).
4. Felder, R.M., and Silverman, L.K., Learning and Teaching Styles in Engineering Education, *Engineering Education*, 78 (7), 674-681 (1988).
5. Montgomery, S., Addressing Diverse Learning Styles Through the Use of Multimedia. *1995 Frontiers in Education Conference Proceedings*, http://fre.www.ecn.purdue.edu/fre/asee/fie95/3a2/3a22/3a22.htm (1995).
6. Doering, E.R., Circuit*Viz*: A New Model for Visualizing the Dynamic Behavior of Electric Circuits. *IEEE Transactions on Education*, 39 (3), 297-303 (1996).
7. Lee, P.M. and Sullivan, W.G., Developing and Implementing Interactive Multimedia in Education. *IEEE Transactions on Education*, 39 (3), 430-435 (1996).
8. Penfield, P., and Larson, R.C., Education Via Advanced Technologies. *IEEE Transactions on Education*, 39 (3), 436-443 (1996).
9. Sears, A.L., and Watkins, S.E., A Multimedia Manual on the World Wide Web for Telecommunications Equipment. *IEEE Transactions on Education*, 39 (3), 342-348 (1996).
10. Butz, B.P., Interactive Multimedia Software for a Course in Introductory Circuits. *Proceedings of the AACE World Conference on Educational Multimedia and Hypermedia*, Calgary, Alberta, Canada (1997).
11. Mosterman, P.J. *et al*, Design and Implementation of an Electronics Laboratory Simulator. *IEEE Transactions on Education*, 39 (3), 327-335 (1996).
12. Harger, R.O., Teaching in a Computer Classroom with a Hyperlinked Interactive Book. *IEEE Transactions on Education*, 39 (3), 327-335 (1996).

13. Barrier Free Education Program, Georgia Institute of Technology Web Site http://barrier-free.arch.gatech.edu/Lab/, (1999).
14. Loyd, D.B., Phalangas, A., and Barner, K.E., An Audio and Speech Based Interface for Computer Controlled Scientific Instruments, *IEEE Transactions on Rehabilitation Engineering* in press (1999).
15. Milchus, K., Using Computer-Controlled Laboratory Equipment To Design Accessible Experiments. Technology and Persons with Disabilities Proc. 1998 CSUN Conference, Los Angeles, http://www.dinf.org/csun_98/csun98_145.htm, (1999).
16. Woitena, G., and Parkes, K., Distance Learning Industry Survey. http://lonestar.texas.net/~gwoitena/distance_ed/sec5/sec5p2.html, (1996).
17. Bourne, J.R. *et al*, Establishing a Web Resource and On-Line Journal for Asnychronous Learning Networks. *Journal of Asynchronous Learning Networks* 1 (1), http://jrbnt.vuse.vanderbilt.edu/alnsociety/backgrou.htm, (1996).
18. Wright, M., *Computers in Musical Applications*, An On-Line Course, http://www.oll.temple/edu/MUS-ST-315/, (1996).
19. Edminister, J.A., *Electric Circuits*, Schaum's Interactive Online Series, New York, NY, McGraw-Hill, Inc, (1995).
20. DiStefano, J.J., Stubberud, A.R., and Williams, I.J., *Feedback and Control Systems*, Schaum's Interactive Outline Series, New York, NY, McGraw-Hill, Inc, (1995).
21. Quinney, D., Harding, R., and IntelliPro, Inc., *Calculus Connections A Multimedia Adventure*, New York, NY, John Wiley and Sons, Inc, (1996).
22. Iskander, M.F. *et al*, Interactive Multimedia for Education. *Proceedings of the 1995 Frontiers in Education Conference*, http://fairway.ecn.purdue.edu/fre/asee/fie95/3a2/3a21/3a21htm, (1995).
23. Barr, A., and Feigenbaum, E., *The Handbook of Artificial Intelligence*, Volume 2, Chapter 9, Reading, MA, Addison-Wesley Publishing Company, Inc. (1982).
24. Sleeman, D., and Brown, J.S. editors, *Intelligent Tutoring Systems*, New York, NY, Academic Press (1982).
25. Brown, J.S., and Burton, R.R., Diagnostic Models for Procedural Bugs in Basic Mathematical Skills. *Cognitive Science* 2 155-192 (1978).
26. Brown, J.S., Burton, R.R., and Larkin, K.M., Representing and Using Procedural Bugs for Educational Purposes. *Proceedings of the 1977 Annual Conference of the ACM*, Seattle, WA, 247-255 (1977).
27. Blaine, L.H., and Smith, R.L., Intelligent CAI: The Role of Curriculum in Suggesting Computational Models of Reasoning. *Proceedings of the 1977 Annual Conference of the ACM*, Seattle, WA (1977).
28.] Brown, J.S., Rubinstein, R., and Burton, R., Reactive Learning Environment for Computer Assisted Electronics Instruction. BBN Report Number 3314, Bolt, Beranek and Newman, Inc., Cambridge, MA (1977).
29. Clancey, W.J. , Dialogue Management for Rule-Based Tutorials. *Proceedings of the Sixth International Joint Conferences on Artificial Intelligence*, 155-161 (1979).
30. Stevens, A.L., *et al*, Quantitative and Qualitative Simulation in Computer-Based Training. *Journal of Computer-Based Instruction*, 10 (1-2) 16-19 (1983).
31. Harmon, P. and King, D., *Expert Systems*, New York, NY, John Wiley and Sons, Inc (1985).
32. Wenger, E., *Artificial Intelligence and Tutoring Systems*. Morgan Kaufmann (1987).
33. Polson, M and Richardson, J.J., editors , *Foundations of Intelligent Tutoring Systems*. New York, NY, Addison Wesley (1988).
34. Van Lehn, K., Conceptual and MetaLearning During Coached Problem Solving. *Proceedings of ITS-96* (1996).
35. Macromedia, *Using Authorware - Version 3*, San Francisco, CA, Macromedia, Inc (1995).
36.] Butz, B.P., Expert Systems. *Abacus*, 5 (1) 30-44 (1987).
37. Giarratano, J.C. and Riley, G.D., *Expert Systems: Principles and Programming*, 3rd edition, Boston, MA, PWS Publishing (1998).
38. Giarratano, J.C., *CLIPS Users Guide*. Artificial Intelligence Section, Lyndon B. Johnson Space Center (1989).

BRIAN P. BUTZ
Brian P. Butz is a Professor of Elecrical Engineering at Temple University, Philadelphia, PA. He founded and is the Director of the Intelligent Systems Application Center. At Temple, he is the co-chair of the University's Distance learning Committee and is a founding member of its Teaching, Learning and Technology Roundtable. His research is in expert systems and intelligent tutoring systems. He is a senior member of the IEEE and the Chair-elect of the IEEE's Philadelphia Section. He received his PhD in electrical engineering from Drexel University, Philadelphia, PA.