

Intelligent and Interactive Web-Based Tutoring System in Engineering Education: Reviews, Perspectives and Development

Arun S. Patil¹ and Ajith Abraham²

¹CQUniversity Mackay Campus, Australia
a.patil@cqu.edu.au

²Machine Intelligence Research Labs (MIR Labs)
Scientific Network for Innovation and Research Excellence
P.O. Box 2259
Auburn, Washington 98071-2259, USA
ajith.abraham@ieee.org
<http://www.mirlabs.org>
<http://www.softcomputing.net>

Abstract. Due to the rapid growth of the use of computers and increasing use of the Internet in education a large number of Web-based educational applications have been developed and implemented. However, very few of them are pedagogically intelligent and interactive for learning purposes. The Web-based intelligent learning has become more effective in the past decade due to increasing use of the Internet in education. A literature search indicates that there is a lack of relevant comprehensive research concerning the efficiency of computer-assisted instructions used in engineering education. The main focus of the research described in this chapter is on the comprehensive review of design and development of the Web-based authoring tool for an Intelligent Tutoring System in engineering education. The chapter outlines and discusses important issues of the development of Intelligent Tutoring System (ITS) in engineering education with an example of the development of a Web-Based Computer-Assisted Tutorials and Laboratory Procedures (WCALP).

1 Computers in Education

The computer-assisted instruction (CAI) in education effectively began in the late 1950s with the use of time-sharing computers, where communication was mainly through the mainframe computers. Hativa (1989, p. 81) suggested that ... *the computer-assisted instruction (CAI) systems designed for the same educational objectives may be based on entirely different concepts and methods of operations and may produce different students attitudes and achievements.* According to Atkinson and Wilson (1969), the three important factors that contribute heavily to the growth of computer-assisted instruction are:

- Development of programmed instruction;
- Mushrooming of electronic data processing;
- Increasing aid to education sectors.

Kulik's research in the 1980s was focused on the use of computer-mediated instruction and found positive research outcomes. The meta-analysis, conducted on 254 controlled evaluation studies showed positive learning effect of computer-based instruction (Kulik & Kulik, 1991). Kulik's meta-analysis studies were carried out when software tutorial programs had limited options. Nevertheless, Kulik's work has influenced educationalists and developers of computer-based instruction authoring systems for the last two decades. In an interesting study into assessing the effects of simulations and higher order thinking technologies with over 13,000 fourth and eighth grade students, Wenglinsky found gains in students' mathematics scores (Schacter, 2003). Also, the higher order use of computers was affected positively on most students' academic achievements. According to Pudlowski (1995), the use of computers in the teaching process should be considered as a supplement to available methods of instruction, rather than the universal remedy for all of the deficiencies that occur in the computer-based education.

The three most common modes of CAI used in learning are:

- Tutorials;
- Drills and practices;
- Interactive simulations.

1.1 Advantages of CAI

Computer-assisted instruction plays a very important role in the modern education process and extensive research shows increasing evidence of the use of computers in the teaching-learning process. There is a body of evidence that computer-assisted instruction, which focuses on higher-order learning in technical education, has been more effective than traditional instruction (Yaakub, 2001). It has been shown that computer-assisted instruction has many advantages in teaching-learning processes, including the following important elements:

- Learners can progress at their own rate and pace.
- Learners can have more individual considerations.
- Immediate feedback is available.
- Learner's response can be recorded.
- Visualisation is properly increased.
- Remedial teaching is eliminated and the problem of discontinuance of learning is avoided.
- A deeper analysis of transfer phenomena is permitted and facilitated.
- An opportunity for the introduction of more effective methods for testing in the teaching/learning process is provided.
- Qualitative and quantitative analysis of student achievements is facilitated.
- Access to supplementary material through links is provided.
- Several complex process and interactions can be explained in simple format.

However, comprehensive research is required to determine the best methodology to be applied to the design and development of computer-assisted instruction, as well as the efficiency of the teaching/learning processes based on this particular method of instruction (Patil, 2004).

2 Computer Assisted Authoring in Engineering Education

There has been significant growth in the use of computers in engineering education; this varies from classroom learning to remote wireless access distance learning. The most important and core part of computer-assisted instruction is to devise and design teaching algorithms. In order to develop any computer-assisted authoring program, two important components to be considered by the developers are:

1. High-performance software;
2. Accessibility.

Advancements in sophisticated computer software tools and hardware technology have resulted in significant advances in courseware authoring tools. The use of modelling and simulation techniques in engineering education is becoming increasingly common over recent years. Early CAI programs were very simple with minimal interaction required. However the use of spreadsheets expanded considerably in the late 1980s and, since then, they are commonly used in almost all engineering disciplines. Advantages of using spreadsheet include the fact that they are easy to learn, easy to debug, and have appropriate graphic and scientific functions. Their excellent problem-solving speed made it especially popular in engineering education (Wankat & Oreovicz, 1993). The development of suitable software was somewhat complex due to the high degree of iteration, as well as problems with debugging. As a result, the availability of authoring tools was limited due to constraints with hardware platforms (Toogod & Wong, 1993).

Since engineering education involves calculations and problem solving, symbolic algebra programs were designed and developed, such as *Mathematica*, *Derive*, *Mapple*, etc. However, in the last decade, parallel development had been evidenced with regard to hardware and software. This has occurred at an incredible rate and resulted in the development of commercial software applications like *SPICE*, *pSPICE*, *ASPEN*, *LabVIEW*, *Labtech*, *WebCT*, *ToolBook* and various CAD programs. Considering the common use of computer operating systems, such as Microsoft Windows, dedicated CAI authoring packages like *Authorware Professional for Windows* have been developed as new generation software packages (Toogood & Wong, 1993). However, the various easily available commercial packages available in the marketplace can be very complex to use, expensive and lack those functions that are considered important for certain learning tasks (Ertugrul, 2000).

3 Intelligent Tutoring Systems (ITS)

An Intelligent Tutoring Systems (ITS) has its roots in the generative Computer-assisted instruction (CAI) and is often known as Intelligent Computer-Assisted Instruction (ICAI) (Sleeman & Brown, 1982). Computer-aided instruction were the forbears to ITS, which started in the late 1950s. The more sophisticated tutoring systems were developed due to the common use of artificial intelligence, which basically started in the 1970s. The intelligent CAI system has the potential

to provide the necessary interaction to suit various learners, and differentiates clearly between Computer-Assisted Instruction (CAI) and Intelligent Computer-assisted instruction (ICAI) (Boulay, 1998). Table 1 briefly lists the main differences between CAI and Intelligent CAI.

Table 1. Guide to differentiating intelligent CAI systems from CAI systems

Computer-assisted instructional systems (CAI)	Intelligent computer-assisted instructional systems (ICAI)
Knowledgeable Can be intelligent Slow learning Tutor controlled Active feedback	Highly knowledgeable Maximum intelligence Fast learning Student controlled Interactive feedback

The Intelligent Tutoring System is one of the fast developing and most popular areas of Computer-Assisted Learning (CAL). According to Ong and Ramachandran (2000), the beauty of ITS technology is that it provides learners with highly interactive learning environments that enable students to practice their skills by carrying out learning tasks. The use of intelligent machines for teaching in education has been around since 1926, when Sidney L. Pressey built a machine with multiple choice questions and answers, which also provided immediate feedback to the user (Thomas, 2003). The three essential components or models of ITS are as follows:

- Knowledge of the domain or expert model;
- Knowledge of the learner or student model;
- Knowledge of teacher strategies or instructor model (Thomas, 2003, Ong & Ramachandran, 2000).

3.1 Artificial Intelligence, Expert Systems and Simulations

The principle of AI made computers more useful, as well as intelligent, in order to utilize them in all the fields of human life. The application of AI principles is the next advanced step to a Web-based ITS, which began in the 1970s and 1980s. Since then, the influence of AI on software technology has considerably increased. As a result, the use of AI techniques in teaching/learning, such as expert systems, simulations and robotics, etc, has become a major factor in the development of Web-based intelligent authoring systems. AI is an advanced scientific technology that is used for efficient computer-based problem-solving techniques in various disciplines. The literature survey shows that the advancement in hardware and software systems has resulted in the development of numerous AI-based authoring tools with the help of sophisticated programming languages, such as LISP or Prolog. There are numerous interactive simulation tools developed in basic engineering disciplines, for example: a prototype of an intelligent case-based process planning system (Yang & Lu, 1993) and three-dimensional mechanical assemblies (Fang & Liou, 1993). However, most of these simulation systems are standalone in nature.

The important contribution of AI in computer-based education is to provide knowledge-based access to resources. Wilson and Welsh (1991) divided AI into three broad areas where knowledge-based systems or expert systems can have important implications for education and training. The history of computerised educational measurement system shows that each generation of educational measurement has shown an increased use of AI and expert systems approaches in order to improve educational measurement activities (Olsen, 1991). The four important generations highlighted by Olsen (1991) are:

- Computerised testing;
- Computerised adaptive testing;
- Continuous measurement;
- Intelligent measurement.

In engineering, science and technology subjects, most of the concepts comprise of mathematical calculations, complex phenomena and graphical representations. Students need to practise in order to familiarise and understand the concepts. Simulations and computer modelling are the most suitable techniques, which provide the proper learning environment. Wedekind (1988) outlined the common structure of all computer simulation programs used in computer-assisted learning and according to him computer simulation programs include input and output routines, plus numerical and graphical routines. ITSs also have the advantages of incorporating simulations, which help in exercising and enhancing learners' expertise in specific areas. SOPHIE, QUEST and STEAMER projects are a few examples of simulation-based tutoring systems. However, the special purposes simulation-based tutoring projects were developed in the late 1970s and 1980s. The sophisticated authoring systems such as; DIAG, RIDES, VIVIDS, etc, developed in the 1990s, provide essential graphical tools, so that students can interact with the software and use a variety of simulation techniques. These simulation techniques are easy to modify and maintain. An authoring system such as RIDES can also automatically generate instruction, which may include feedback and evaluation (Murray, 1997).

3.2 Working of an ITS

The three major components of any ITS coordinate and work together in order to produce better learning outcomes. These components are listed as below:

- Teacher strategies;
- Learner knowledge;
- Domain knowledge.

Although each component has its specific identity and functioning ability, they always interact and coordinate with each other that help in guiding learners and interfacing them to expert in the subject. In a typical formal teaching situation, guidance is provided based on the learner's performance observed and gained during the instruction. However, in ITS-based learning, highly interactive techniques, such as simulations, AI and expert systems are incorporated with the learning in order to perform the action. The information received by the learner is again used

as a feedback to provide further instructions and the cycle is repeated. Beck et al (1996) categorised ITSs on two dimensions, namely: abstraction of the learning environment and the knowledge type of the instruction.

3.3 ITS and World Wide Web

Most of the traditionally developed ITSs use static media, such as CD-ROMs. However, it is very difficult to deploy the advancement and changes in the courseware in such systems; hence it is generally not suitable for basic learning strategies. The way to minimise the disadvantages inherent in traditional Intelligent Tutoring Systems is to utilise World Wide Web (WWW). Since its inception in 1990, the Internet (WWW or Web) has quickly emerged as a powerful new tool for connecting people and information on a global scale (Reed & Afjeh, 1998). The Web technology can also be utilised for various learning programs based on multimedia/hypermedia techniques. ITS developers mostly concentrate on designing a program that can provide effective instructions for the desired learning task. The developer's task has become more advanced and useful but challenging due to the common use of Internet technology in educational instructional systems. Recently, several advanced ITSs have been used for open and distance learning in many universities that provide learners with adequate sources of learning material for unlimited, unconditional use via the Internet.

4 ITS Developments

In the initial phase of ITS developments, several efforts have been undertaken to develop tutoring systems, mostly for teaching computer programming to students at the university level (Larkin and Chabay, 1992). However, recent research on ITS developments indicates that there are few dedicated ITSs developed in the engineering domain. Dear (1991) identified the trend in developing computer-assisted instruction regarding the relationship of the content and the code. According to him, in traditional authoring systems developed with general purpose programming languages (eg BASIC, COBOL, PASCAL, etc), the code and the contents were one and the same, yet this trend is changing with regard to AI-based authoring systems in which content is being separated from the code.

Curilem et al (2006) propose a mathematical model of Intelligent Tutoring Systems (ITS), based on observations of the behaviour of these systems. One of the most important problems of pedagogical software is to establish a common language between the knowledge areas involved in their development, basically pedagogical, computing and domain areas. Authors presented an example demonstrating how the formalization was used to design the adaptive mechanism of and to adapt its Interface Module to some student characteristics.

Crowley and Medvedeva (2006) described the development of a general intelligent tutoring system for teaching visual classification problem solving. The architecture incorporates aspects of cognitive tutoring system and knowledge-based system design within the framework of the unified problem-solving method

description language component model. Based on the domain ontology, domain task ontology and case data, the abstract problem-solving methods of the expert model create a dynamic solution graph. Student interaction with the solution graph is filtered through an instructional layer, which is created by a second set of abstract problem-solving methods and pedagogic ontologies, in response to the current state of the student model.

Huang et al. (2006) proposed three kinds of learning parameter improvement mechanisms to establish effective parameters that are frequently used in the learning platforms. The proposed learning parameter improvement mechanisms can calculate the students' effective online learning time, extract the portion of a message in discussion section which is strongly related to the learning topics, and detect plagiarism in students' homework, respectively. The derived numeric parameters are then fed into a Support Vector Machine (SVM) classifier to predict each learner's performance in order to verify whether they mirror the student's studying behaviors. The experimental results show that the prediction rate for the SVM classifier can be increased up to 35.7% in average after the inputs to the classifier are "purified" by the learning parameter improvement mechanisms.

Woo et al. (2006) main research objective was to build an intelligent tutoring system capable of carrying on a natural language dialogue with a student who is solving a problem in physiology. Analysis of a corpus of 75 hour-long tutoring sessions carried on in keyboard-to-keyboard style by two professors of physiology tutoring first-year medical students provided the rules used in tutoring strategies and tactics, parsing, and text generation. The system presents the student with a perturbation to the blood pressure, asks for qualitative predictions of the changes produced in seven important cardiovascular variables, and then launches a dialogue to correct any errors and to probe for possible misconceptions. The natural language understanding component uses a cascade of finite-state machines. Results of experiments with pretests and posttests have shown that using the system for an hour produces significant learning gains and also that even this brief use improves the student's ability to solve problems more than reading textual material on the topic. Han et al. (2005) described an intelligent tutoring in a collaborative medical tutor for Problem Based Learning (PBL). Authors used Bayesian networks to model individual student clinical reasoning, as well as that of the group. The prototype system incorporates substantial domain knowledge in the areas of head injury, stroke and heart attack. Tutoring in PBL is particularly challenging since the tutor should provide as little guidance as possible while at the same time not allowing the students to get lost. From studies of PBL sessions at a local medical school, the authors identified and implemented eight commonly used hinting strategies.

Stathacopoulou et al. (2005) proposed a neural network implementation for a fuzzy logic-based model of the diagnostic process is proposed as a means to achieve accurate student diagnosis and updates of the student model in Intelligent Learning Environments. The neuro-fuzzy synergy allows the diagnostic model to some extent "imitate" teachers in diagnosing students' characteristics, and equips the intelligent learning environment with reasoning capabilities that can be further used to drive pedagogical decisions depending on the student learning style. The

neuro-fuzzy implementation helps to encode both structured and non-structured teachers' knowledge: when teachers' reasoning is available and well defined, it can be encoded in the form of fuzzy rules; when teachers' reasoning is not well defined but is available through practical examples illustrating their experience, then the networks can be trained to represent this experience. Jaques and Vicari (2006) described the use of mental states approach, more specifically the belief-desire-intention (BDI) model, to implement the process of affective diagnosis in an educational environment. Authors used the psychological OCC model, which is based on the cognitive theory of emotions and is possible to be implemented computationally, in order to infer the learner's emotions from his actions in the system interface. The proposed architecture profit from the reasoning capacity of the BDI model in order to infer the student's appraisal (a cognitive evaluation of a person that elicits an emotion) which allows to deduce student's emotions. The system reasons about an emotion-generating situation and tries to infer the user's emotion by using the OCC model. Besides, the BDI model is very adequate to infer and also model student's affective states since the emotions have a dynamic nature.

Aberek and Popov (2004) presented an intelligent tutoring system STATFAG for education on design, optimisation and the manufacturing of gears and gearing. STATFAG serves several purposes. It helps lecturers to plan and execute optimal lessons for various subjects; it makes self-studies of students of higher level of education at the Faculty of Mechanical Engineering much more efficient; and also assists less experienced graduates who must independently design gear assemblies in companies. Hwang (2003) proposed a conceptual map model, which provides learning suggestions by analyzing the subject materials and test results. A testing and diagnostic system is also implemented on computer networks based on the novel approach. Experimental results have demonstrated that the novel approach benefits students and deserves further investigation.

5 A Case Study of WCALP

Woolf (1992) claims that several iterative cycles are required to develop an intelligent tutor or any AI system. It is very hard to design and develop an ideal ITS that may cater all pedagogical needs and also will suit with available software technologies. However, while developing any ITS developers must consider two important levels of integration: one from the developer's point of view and the other from the learner's point of view. Beck et al (1996) suggested a strategy for the design of effective ITSs with five essential components, namely: the student model, the pedagogical model, the domain knowledge, the communications model and the expert model. Their research on the applications of AI in education added the *expert model* as the fifth component to the four major components already identified by Woolf. Furthermore, they pointed out the importance of a large research potential in multiple teaching strategies in the pedagogical model. Table 2 gives a list of essential points to be considered to formulate the strategy in developing the Web-based educational system or a Web-based ITS.

Table 2. Essential points to be considered in designing an ITS strategy (Patil, 2004)

Developer's View	Learner's (End User's) View
<ul style="list-style-type: none"> • Course material • Systems available • Software design • Accessibility • Learning pedagogy • Learning outcomes • Cost-benefits 	<ul style="list-style-type: none"> • Learning approach • Change of behaviour • Metacognition • Time and usability • Cost-benefits • Assessments and evaluations

5.1 Methodology

The development of Web-based ITS have two important levels of integration: one from the developer's point of view and the other from the learner's point of view. To develop any Web-based ITS is a tedious task. Moreover, the developers must keep in mind that the Web-based educational systems should have the following basic characteristics:

1. Dynamic;
2. Interactive;
3. Intelligent;
4. Adaptive;
5. Flexible;
6. Learner centred (Patil, 2004).

It is important to have proper organisation of knowledge or curriculum to be taught in designing the Web-based ITS. Albacete and VanLehn (2000) identify the manner in which the desired knowledge is presented is very important factor in human tutoring techniques. Nevertheless, CAI, or Web-based learning, is an evolutionary process and has its own unique characteristics regarding the choice of appropriate pedagogy. In designing the project pedagogy, the following factors will be taken into consideration:

- Content of the course
- Learner's activities
- Mode and level of interaction
- Impact of new approach
- Performance indicator
- Learning outcomes.

5.2 System Architecture

Any computer-assisted learning or ITS must have essential courseware design that can easily fulfil the important pedagogical requirements in the learning strategy. It

has been proved with much research in the area of online learning that an integral approach to online learning can be used to promote students' critical use, understanding and application of materials. The proposed system architecture of a Web-based pedagogical ITS module can be explained with a block diagram shown in Figure 1. The most important part of such an architecture is a domain layer with learners' interaction within the pedagogical control unit and their responses. It is also possible to enhance the pedagogical aspects in the teaching-learning environment by incorporating pedagogical agents into the design in order to make the tutorials more effective.

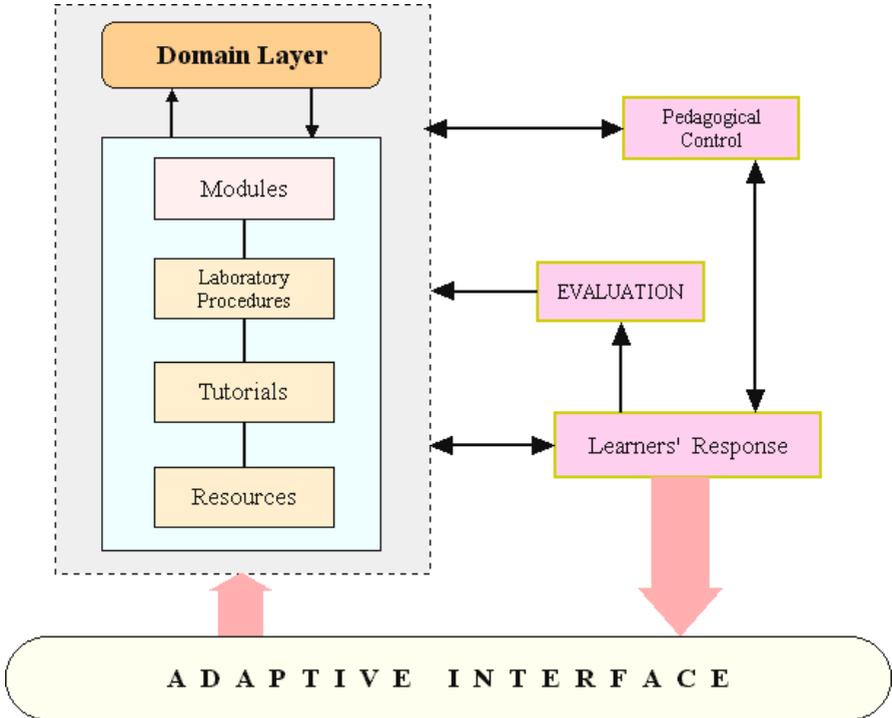


Fig. 1. System architecture of a pedagogical ITS module (Patil, 2004)

The system architecture used in the Web-Based Computer-Assisted Tutorials and Laboratory Procedures (WCALP) is based on *client-server* configuration. In this design, the user or client communicates with the server using a Web browser. This architecture has four important components, as outlined in the Figure 2:

- Server-side configuration;
- Library of software and database;
- Client-side configuration;
- User interface.

The user can access WCALP using any browser from his/her machine and perform the tutorial task including laboratory procedures. The responses from the user are returned to the WCALP server for further processing. The results will be calculated from the data collected and will then be analysed and sent back to the user's machine in order to check his/her performance. It is essential to provide the learners in any Web-based ITS with proper motivation for learning, the idea to recall the previous concepts learned, direction for essential immediate feedback and testing the performance. In order to achieve better learning outcomes, the sequence of the learning material can be managed and executed as per the hierarchy given by the flowchart developed.

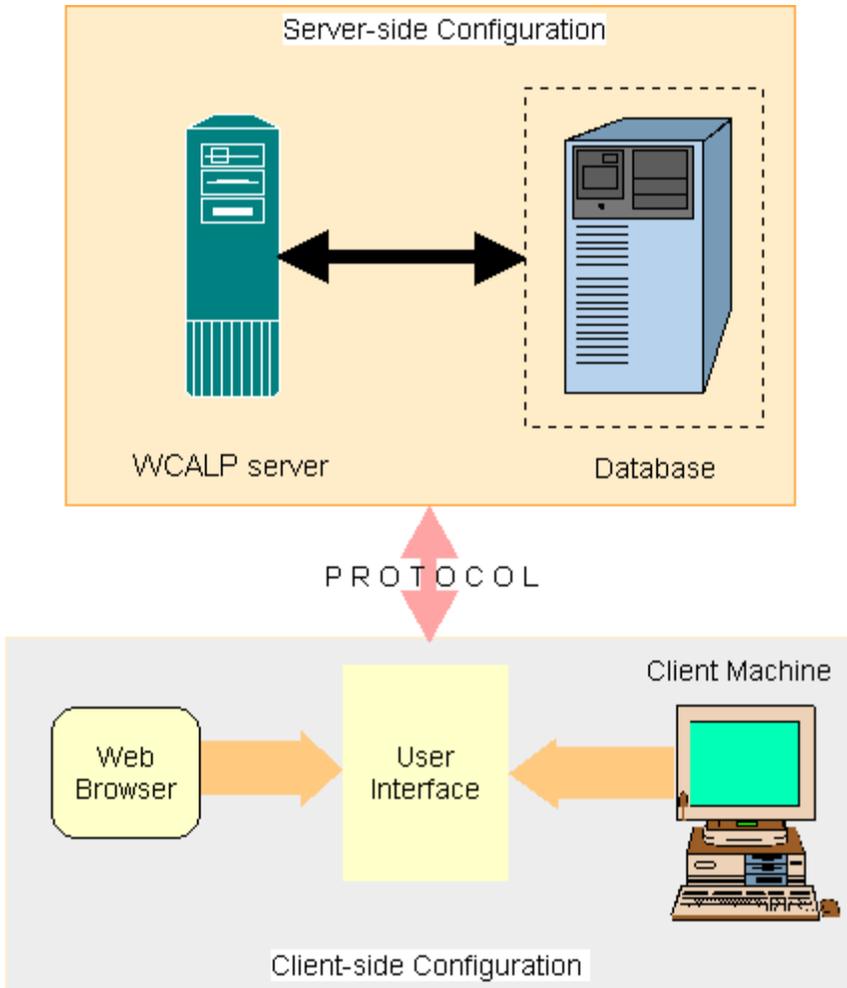


Fig. 2. System configuration of WCALP (Patil, 2004)

5.3 *Authoring Software*

Considering the applicability of the specialised programming language, such as Java (which contains multi-level function performing tasks), it is used as an authoring language to develop and design this project. Java is a fully object-oriented programming language and one of the fastest growing programming technologies of all time. Java represents an evolutionary change from earlier programming languages, rather than a radical departure. It is possible to make a Web page come alive with animations and user interactivity with Java. Java is the most important language for Web and Internet applications.

The WCALP consists of Java applets and Java script programs that are incorporated in the main page of each unit. This provides several exercises that are related to that unit and is useful in order to refresh and recall the necessary information of that particular unit. This type of design basically works with Transmission Control Protocol/Internet Protocol (TCP/IP). All Java applets are activated and opened on the user's (student's) machine with the help of the Java Virtual Machine (JVM) in the Web browser, which actually downloads the programming code and executes it there. As a result, a pop-up window containing an applet is opened that directly communicates with the server.

Although Java applets can provide strong client-side control, there are concerns related to the download and storage of Java applets in case when the file size is comparatively larger. Furthermore, applet functionality can vary between different browsers (Berntzen et al, 2001). The basic problems identified by Junglas (2003) with the developments of Java applets concern a huge amount of work involved and the requirement of profound Java programming knowledge. Java applets are truly platform independent and have in-built security against corrupt or bad coding of the system. However, running multiple and large files of Java applets on a small system can cause loading problems. In order to minimise the difficulties associated with applet configurations, only one or two applets are developed for each Unit, having a small file size that can be loaded easily and run on any system. The most relevant alternative to applets is to use JavaScript.

5.4 *Immediate Feedback*

The tutorial strategies also include the provision of immediate feedback. The central idea of learner interface employed in this project is to record the learner's response into the server database, to provide immediate access to the correct answers or hints and also to provide detailed information about the learner's performance whenever required. This has been achieved by using simple scripting in the Web pages. For example, the *check answer* button in every question window has a link to the correct answer, which allows the user to check the correct answer with a pop-up answer window of that question. The simple script used for this is given in Appendix V.

5.5 CGI Programming and Perl

In the case of the WCALP, it is proposed to use an *MS Access* database where all question and answer files will be stored at the WCALP server. Using query generated by Perl scripting language, the questions can be accessed from the server and responses from the user can be stored back where they are processed and again sent back to the user to check on his/her performance. Although, CGI applications have a better security level, speed is the main concern associated with the CGI programming in the case of a large database, which has to be processed in order to give updated results to the user. Also, CGI scripts can only handle small applications.

6 Issues in Developing ITS and Intelligent CAI

Web-based education is being considered as the replacement for traditional classroom-based education at the tertiary level. Barretto et al (2003) claim that user activity and graphic design are two major attributes that influence the development of educational software projects. Although Web-based instructions have distinct advantages over traditional classroom-based learning, the design and development of appropriate Web-based instructional software is a great challenge. CAL or Web-based learning tools offer more dynamic representations of learning materials than traditional blackboard teaching. Although developing any Web-based educational system involves a series of challenges and issues, three major factors of the development that must be considered are namely:

- The development of instructional software;
- The development of the database management;
- The development of assessment tools.

The literature research on ITS shows that ITSs have not been completely successful when implemented on a larger scale, due to various problems associated with their developments. McArthur et al (1993) pointed out that ITS limitations are influenced by certain components, such as the expert system, pedagogical component, student model and interface. Kinshuk & Patel (1997) claim that the development of ITSs has suffered due to the lack of software development that can be specifically used for tutoring purposes. The important issues incurred during the development of WCALP are discussed below.

6.1 Pedagogical Concerns

ITS has many advantages given the recent advancements in systems of education, including online learning. However, there are few minor concerns and issues related to the development of ITS or Intelligent Computer-Assisted Instruction. Most of ITS developers lack understanding of the implementation of pedagogical concepts that need to be incorporated within the ITS; consequently, the end-product design may not fulfil all of the pedagogical elements required in the tutoring strategies.

6.2 Software Latency

Hardware and software or programming languages are other important concerns associated with AI-based ITS. There are several advanced computer programming languages currently available, yet the application of proper computer language is one of the essential needs in developing intelligent tutoring. Kinshuk & Patel (1997) claim that the development of ITSs has suffered due to the lack of development of underlying software engineering methodologies for tutoring purposes. This gives rise to a need for generic software that can be rapidly customised as per individual needs. Also, the important concern is the design and maintenance of the large database used in these systems. Since ITS or AI-based tutoring consist of several files of information in the form of text, graphics and audio-visual data, it is somewhat cumbersome to manage all of this data. Also, the research shows that in some of the ITSs, debugging is the main concern in the designed software; it is very difficult to debug problems encountered in authoring programs due to large file sizes and complicated interfaces.

6.3 Courseware Development

Any computer-assisted learning or ITS must have essential courseware design that can easily fulfil the required pedagogical requirements in the learning strategy. In designing the project pedagogy for WCALP, several important factors have been taken into consideration, such as: the content of the course, learner's activities, performance indicators and learning outcomes. The WCALP has been developed using a framework in which three key strategies have been embedded in the model, namely: the tutorial strategy, hint strategy and evaluation strategy. This framework is very much pertinent in the development of Web-based educational software to help teaching/learning.

6.4 Authoring Software

The instructional software designed has the following important characteristics that are essential in order to implement the developed system with the appropriate instructional strategies. These characteristics are pertinent in the design of ideal instructional software for Web-based education in the engineering domain. These characteristics are listed as follows:

- The software is user friendly, easy to use and task-oriented.
- The instructional design strategies used in the software are able to provide the most appropriate encouragement for self-motivation.
- The software incorporates simple Web delivery methods with the minimum standards required of the desired content.
- The developed software provides realistic solutions to the problems according to the complex to simple principle.

- The software is learner-centred and not developer-centred, which provides opportunities for learners to test theories and explore their own learning.
- The software also develops active learning and facilitates essential thought processes, which result in a positive impact on the learning outcomes.
- The software utilises appropriate technology and includes educational facilities for self-assessment by learners throughout the learning process (Patil, 2004).

7 Conclusions and Future Work

The Web-based Intelligent Tutoring Systems (ITSs) are developing rapidly since the inception of the World Wide Web; however, very few of these are dedicatedly designed and developed for engineering domain so far. In this chapter, we described the methodology on the effectiveness of the developed Web-based intelligent and interactive systems in order to test its potential in learning pedagogy. The novel learning pedagogy applied in the circuit theory for basic electrical engineering has been also explained with the developed interactive Web pages. The next major goal for this developed system is to research on its effectiveness by data collection and analysis. This research can also be extended to develop and evaluate the Web-based model for other engineering related domains in the future.

7.1 Software Updating

The language used in the development of WCALP is Java in order to incorporate interactive tutorial components. Java programming has the advantage of an object oriented programming facility that supports several types of Web developments. However, the use of Java applets in Web applications has been limited due to its downloading problems and functionality with different browsers. As a result, it is envisaged that either advanced JSP or Java Servlet technology.

7.2 Using Advanced Java Solutions

With regard to JSP or Java Servlets, a dedicated server will be used, such as Apache Web server or Apache JServ servlet server. This type of server configuration and a combination of Java with other application-programming interface can provide efficient multi-user, real time interaction (Chang & Hung, 2000). This will help in enabling the use of several Web server systems in various formats like HTML, JavaScripts etc.

7.3 Using Commercial Software

Recent developments in computer-assisted instruction have shown that several interactive authoring solutions for computer-delivered control, simulation and scientific visualisations are available in the market (Pudlowski, 1995b). Evidence

has also been found of similar developments that have been generated with the aid of several types of commercial authoring software, such as *Authorware*, *Director*, *LabVIEW*, *HyperText*, *WebCT*, etc. Recent developments of LabVIEW-based applications have shown that many classroom-based engineering courses and conventional laboratory work can be combined with custom-written virtual instrumentation by computer-assisted delivery (Ertugrul, 2000).

Another option in the future developments of the WCALP is to utilise and test the *LabVIEW* package for advanced laboratory simulations and experiments in circuit theory. Since several commercial authoring software packages available are very complex to use, expensive and lack all of the functions required for certain learning tasks (Ertugrul, 2000), the possible integration of the *LabVIEW* package will be tested as pilot project for its ease of use and effectiveness. The advantage of using these types of readymade software tools is that they provide flexible navigation and other designed tools so that developers can concentrate more on pedagogical design issues, rather than technological aspects.

7.4 Adding Interactive Hypermedia and Animations

In future developments, interactive hypermedia and animations may be added to the WCALP with the integration of Flash technology. In using this combination of Java and Flash, learners can easily interact with the animations and demonstrations in learning theories. Computer-based simulations provide an excellent environment as an advanced teaching tool for several subjects in engineering and technology education (Roorda & Cartwright, 1994, Kanflo & Sagert, 1994, Veith et al, 1998). When applied to engineering education, the benefits of Web-based simulations as pointed out by Veith et al (1998), include controlled access, wide accessibility, increased integration and efficient maintenance. This is possible with the integration of the object oriented programming language like Java. According to Veith et al (1998), Internet-based simulations can help educators incorporate simulation models into their courses without cost, as well as avoiding inconveniences that arise due to the special equipment requirements.

References

1. Aberek, B., Popov, V.: Intelligent tutoring system for training in design and manufacturing. *Advances in Engineering Software* 35(7), 461–471 (2004)
2. Albacete, P.L., VanLehn, K.: The conceptual helper: an intelligent tutoring system for teaching fundamental physics concepts. In: Gauthier, G., VanLehn, K., Frasson, C. (eds.) *ITS 2000. LNCS*, vol. 1839, pp. 564–573. Springer, Heidelberg (2000)
3. Atkinson, R.C., Wilson, H.A. (eds.): *Computer-Assisted Instructions*. Academic Press, New York (1969)
4. Barretto, S.F.A., Piazzalunga, R., Ribeiro, V.G., Dalla, M.B.C., Filho, R.M.L.: Combining interactivity and improvement layout while creating educational software for the Web. *Computers and Education* 40(3), 271–284 (2003)

5. Berntzen, R., Strandman, J.O., Fjeldly, T.A., Shur, M.S.: Advanced solutions for performing real experiments over the Internet. In: International Conference on Engineering Education, Oslo, Norway (2001), <http://www.ineer.org/Events/ICEE2001/Proceedings/Index.htm> (Retrieved November 18, 2003)
6. Boulay, B.d.: What does the “AI” in AIED buy? In: Colloquium on Artificial Intelligence in Educational Software, IEE Digest No: 98/313, pp. 3/1–3/4 (1998)
7. Chang, T., Hung, D.: Web-based distance experiments: design and implementation. In: IEEE International Conference on Engineering Education, Taipei, Taiwan (2000), <http://www.ineer.org/Events/ICEE2000/Proceedings/papers/WA4-4.pdf>
8. Crowley, R.S., Medvedeva, O.: An intelligent tutoring system for visual classification problem solving. *Artificial Intelligence in Medicine* 36(1), 85–117 (2006)
9. Curilem, S.G., Barbosa, A.R., de Azevedo, F.M.: Intelligent tutoring systems: Formalization as automata and interface design using neural networks. In: *Computers and Education* (2006) (in press), <http://dx.doi.org/10.1016/j.compedu.2005.10.005>
10. Dear, B.L.: Artificial Intelligence Techniques: Applications for Courseware Development. In: *Expert Systems and Intelligent Computer-Aided Instructions*, pp. 91–99. Educational Technology Publications, Englewood Cliffs (1991)
11. Ertugrul, N.: Towards virtual laboratories: a survey of *LabVIEW*-based teaching/learning tools and future trends. *International Journal of Engineering Education* 16(3), 171–180 (2000)
12. Fang, Y., Liou, F.W.: Computer simulation of three-dimensional mechanical assemblies: part-i – general formulation. In: 1993 ASME Inter. Computers in Education Conference and Exhibition, pp. 579–587 (1993)
13. Godwin-Jones, B.: Emerging technologies, dynamic Web page creation. *Language, Learning and Technology Journal* 1(2), 7–13 (1998), <http://polyglot.cal.msu.edu/llt/vol1num2/emerging/default.htm> (Retrieved March 12, 2002)
14. Graham, D.T., McNeil, J., Pettiford, L.: *Untangled Web: Developing Teaching on the Internet*. Pearson Education, Harlow (2000)
15. Gwo-Jen Hwang, G.-J.: A conceptual map model for developing intelligent tutoring systems. *Computers and Education* 40(3), 217–235 (2003)
16. Han, S.-G., Lee, S.-G., Jo, G.-S.: Case-based tutoring systems for procedural problem solving on the www. *Expert Systems with Applications* 29(3), 73–82 (2005)
17. Hativa, N.: Students’ conceptions of, and attitudes toward, specific features of a CAI system. *Journal of Computer-Based Instruction* 16(3), 81–89 (1989)
18. Huang, C.-J., Chu, S.-S., Guan, C.-T.: Implementation and performance evaluation of parameter improvement mechanisms for intelligent e-learning systems. *Computers and Education* (2006) (in press), <http://dx.doi.org/10.1016/j.compedu.2005.11.008>
19. Jaques, P.A., Vicari, R.M.: A BDI approach to infer student’s emotions in an intelligent learning environment. *Computers and Education* (2006) (in press), <http://dx.doi.org/10.1016/j.compedu.2005.09.002>
20. Junglas, P.: Using applets for physics education: a case study of non-linear systems and chaos. In: 7th Baltic Region Seminar on Engineering. Education, pp. 61–64 (2003)
21. Kanflo, T.J., Sagert, P.A.: Teaching the design of manufacturing systems using simulation. *European Journal of Engineering Education* 19(1), 117–123 (1994)

22. Kinshuk, Patel, A.: A Conceptual Framework for Internet Based Intelligent Tutoring Systems. In: Behrooz, A. (ed.) Knowledge Transfer, vol. 2, pp. 117–124. pAce, London (1997)
23. Kulik, C.-L., Kulik, J.A.: Effectiveness of computer-based instruction: an updated analysis. *Computers in Human Behaviour* 7(1-2), 75–94 (1991)
24. Larkin, J.H., Chabay, R.W. (eds.): *Computer-Assisted Instruction and Intelligent Tutoring Systems: Shared Goals and Complementary Approaches*, pp. 151–186. Lawrence Erlbaum & Assoc., Hillsdale (1992)
25. Murray, T.: Expanding the knowledge acquisition bottleneck for intelligent tutoring systems. *International Journal of Artificial Intelligence in Education* 8(3) (1997)
26. Olsen, J.B.: The Four Generations of Computerised Testing: toward Increased Use of AI and Expert Systems. In: *Expert Systems and Intelligent Computer-Aided Instructions*, pp. 66–71. Educational Technology Publications, Englewood Cliffs (1991)
27. Ong, J., Ramachandran, S.: Intelligent tutoring systems: the what and the how. *Learning Circuits* (2000), <http://www.learningcircuits.org/feb2000/> (Retrieved June 15, 2003)
28. Patil, A.S.: Issues in the design and development of Web-based computer-assisted tutorials and laboratory procedures in basic electrical engineering. Unpublished masters dissertation, Monash University, Melbourne (2004)
29. Pudlowski, Z.J.: Major Issues in Developing Computer-Aided Authoring Programs for Engineering Education. In: Pudlowski, Z.J., et al. (eds.) *Computers in Electrical Engineering – Research, Development and Application*, pp. 23–31. EEERG & USICEE, Melbourne (1995)
30. Reed, J.A., Afjeh, A.A.: Developing interactive educational software for the World Wide Web with Java. *Computers and Education* 30(3-4), 183–194 (1998)
31. Roorda, J., Cartwright, T.: Engineering design simulation: a computer-based teaching environment. *European Journal of Engineering Education* 19(1), 105–115 (1994)
32. Schacter, J.: The Impact of Education Technology on Student Achievement: What the Most Current Research has to Say (2003), <http://www.mff.org/publications/publications.taf?page=161> (Retrieved October 3, 2003)
33. Sleeman, D., Brown, J.S.: *Intelligent Tutoring Systems*. Academic Press, London (1982)
34. Stathacopoulou, R., Magoulas, G.D., Grigoriadou, M., Samarakou, M.: Neuro-fuzzy knowledge processing in intelligent learning environments for improved student diagnosis. *Information Sciences* 170(2-4, 25), 273–307 (2005)
35. Surber, J., Leeder, J.: The effect of graphic feedback on students' motivation. *Journal of Computer-Based Education* 15(1), 14–17 (1988)
36. Thomas, E.: *Intelligent Tutoring Systems, ITS* (2003), <http://coe.sdsu.edu/eet/Articles/tutoringsystem/start.htm> (Retrieved June 15, 2003)
37. Toogood, R.W., Wong, D.: An interactive fluid mechanics tutorial using authorware. In: 1993 ASME International Computers in Education Conference and Exhibition, pp. 539–545 (1993)
38. Veith, T.L., Kobza, J.E., Koelling, C.P.: World Wide Web-based simulations. *International Journal of Engineering Education* 14(5), 316–321 (1998)
39. Wankat, P.C., Oreovicz, F.S.: *Teaching Engineering*. McGraw-Hill, New York (1993)

40. Wedekind, J.: Computer-Aided Model Building. In: Mandl, H., Lesgold, A. (eds.) *Learning Issues for Intelligent Tutoring Systems*, pp. 287–294. Springer, New York (1988)
41. Wilson, B.G., Welsh, J.R.: *Small Knowledge-Based Systems in Education and Training: Something New under the Sun*. In: *Expert Systems and Intelligent Computer-Aided Instruction*, vol. 2. Educational Technology Publications, Englewood Cliffs (1991)
42. Woo, C.-W., Evens, M.W., Freedman, R., Glass, M., Shim, L.-S., Zhang, Y., Zhou, Y., Michael, J.: An intelligent tutoring system that generates a natural language dialogue using dynamic multi-level planning. In: *Artificial Intelligence in Medicine (2006)* (in press), <http://dx.doi.org/10.1016/j.artmed.2005.10.004>
43. Woolf, B.P.: Building knowledge based tutors. In: Tomek, I. (ed.) *ICCAL 1992*. LNCS, vol. 602, pp. 46–60. Springer, Heidelberg (1992)
44. Yaakub, M.N.: *Effectiveness of Computer-Assisted Instruction in Technical Education: a Meta-Analysis (2001)*, <http://voc.ed.psu.edu/projects/publications/books/Fall%202001/WEF2001.2.html> (Retrieved May 12, 2003)
45. Yang, H., Lu, W.F.: Procace: a prototype of intelligent case-based process planning system with simulation environment. In: *1993 ASME International Computers in Education Conference and Exhibition*, pp. 571–577 (1993)