

2000

Web-based computer assisted laboratory instruction

Yuxin Chen
University of Wollongong

Follow this and additional works at: <https://ro.uow.edu.au/theses>

University of Wollongong

Copyright Warning

You may print or download ONE copy of this document for the purpose of your own research or study. The University does not authorise you to copy, communicate or otherwise make available electronically to any other person any copyright material contained on this site.

You are reminded of the following: This work is copyright. Apart from any use permitted under the Copyright Act 1968, no part of this work may be reproduced by any process, nor may any other exclusive right be exercised, without the permission of the author. Copyright owners are entitled to take legal action against persons who infringe their copyright. A reproduction of material that is protected by copyright may be a copyright infringement. A court may impose penalties and award damages in relation to offences and infringements relating to copyright material.

Higher penalties may apply, and higher damages may be awarded, for offences and infringements involving the conversion of material into digital or electronic form.

Unless otherwise indicated, the views expressed in this thesis are those of the author and do not necessarily represent the views of the University of Wollongong.

Recommended Citation

Chen, Yuxin, Web-based computer assisted laboratory instruction, Master of Engineering (Hons.) thesis, , University of Wollongong, 2000. <https://ro.uow.edu.au/theses/2559>

Research Online is the open access institutional repository for the University of Wollongong. For further information contact the UOW Library: research-pubs@uow.edu.au

Web-Based Computer Assisted Laboratory Instruction

A thesis submitted in fulfilment of the requirements for the award of the degree

Honours Master of Engineering

from

UNIVERSITY OF WOLLONGONG

by

Yuxin Chen

Department of Electrical, Computer & Telecommunications

2000

DECLARATION

I, Yuxin Chen, declare that this thesis, submitted in fulfilment of the requirements for the award of Master of Engineering (Honours), in the Department of Electrical, Computer & Telecommunications, University of Wollongong, is wholly my own work unless otherwise referenced or acknowledged. The document has not been submitted for qualifications at any other academic institution.

Signed,

Yuxin Chen

March, 2000

Acknowledgment

I am very grateful to my supervisor, Associate Professor Fazel Naghdy, for his guidance and financial support. He provided the optimal research environment and advice. I would like to acknowledge his great effort and patience in answering my questions and correcting my errors.

I would also like to thank students in Mechatronics Lab and the staff at the Department of Electrical, Computer & Telecommunications in University of Wollongong for their helps.

Yuxin Chen

Abstract

The feasibility of computer-assisted instruction in a practical laboratory has been explored in this work. Computer assisted instruction (CAI), in which educational instruction is delivered through a computer, has been a popular area of research and development. Computer assisted laboratory instruction (CALI), on the other hand, has not been systematically studied in the past as literature reveals. In the work conducted in this research, the concept of CALI has been examined by developing a web-based multi-media CALI package for Control Systems laboratory that is used by around 100 students annually in the School of Electrical, Computer and Telecommunications Engineering, University of Wollongong. Some elements of Intelligent Tutoring Systems (ITS) have been also incorporated to increase the flexibility of the instruction provided. A systematic approach has been employed to develop the specifications of the package and design its structure to ensure its effectiveness. The latest tools in Web development have been employed to achieve all the defined specifications efficiently and systematically. The outcome is a system that has proved very effective in its operation and instruction for the students in the laboratory. In addition to the specific results and benefits produced directly as the result of employing the package in Control Laboratory, the study has also generated outcomes that are generic and can be considered in the application of the approach in any practical laboratory.

Key Words: Intelligent Tutoring System, multimedia, laboratory instruction, computer-assisted instruction, hypermedia, web-based, Control Systems

List of Abbreviation

ACT	Adaptive Control of Thought
AI	Artificial Intelligence
CaI	Computer-aided Instruction
CAI	Computer-Assisted Instruction
CALI	Computer Assisted Laboratory Instruction
CBI	Computer-Based Instruction
CBT	Computer-Based Training
CEI	Computer-Enriched Instruction
CGI	Common Gateway Interface
CMI	Computer-Managed Instruction
GUI	Graphical User Interface
ICAI	Intelligent Computer-Assisted Instruction
IMTS	Intelligent Multimedia Tutoring System
IP	Internet Protocol
ISDN	Integrated Services Digital Network
ITS	Intelligent Tutoring System
LANs	Local Area Networks
PI	Proportional-plus-Integral
RTSP	Real Time Streaming Protocol
SQL	Structured Query Language
TCL	Traditional Classroom Lecture
URL	Uniform Resource Locator
WWW	World Wide Web

TABLE OF CONTENTS

DECLARATION	I
ACKNOWLEDGMENT	II
ABSTRACT.....	III
LIST OF ABBREVIATION	IV
TABLE OF CONTENTS	V
CHAPTER 1 : INTRODUCTION	1-1
1.1 Project Background.....	1-1
1.2 Shortcomings of Conventional Laboratory Instruction.....	1-2
1.3 Project Motivation and Objectives.....	1-4
1.4 Advantages of CALI	1-6
1.5 ITS Versus CALI	1-7
1.6 Structure of this Thesis.....	1-8
CHAPTER 2 : BACKGROUND	2-1
2.1 Introduction.....	2-1

2.2 Computer Assisted Instruction (CAI)	2-2
2.2.1 Terms Related to CAI.....	2-2
2.2.2 Comparison of CAI and TCL (traditional classroom lecture)	2-3
2.2.3 Web Based CAI	2-5
2.2.4 Evaluating a CAI System.....	2-6
2.2.5 Computer Assisted Laboratory Instruction (CALI).....	2-7
2.3 Evolution from CAI to ITS (Intelligent Tutoring System)	2-9
2.4 Intelligent Tutoring Systems	2-14
2.4.1 Structural Model of ITS.....	2-14
2.4.2 The Expert Module.....	2-15
2.4.3 The Student Module	2-16
2.4.4 The Curriculum and Diagnosis Module.....	2-17
2.4.5 The Interface Module.....	2-18
2.4.6 Some Remarks on ITS	2-18
2.5 Contributing Techniques and Technologies.....	2-19
2.5.1 Multimedia.....	2-19
2.5.2 Hypertext and Hypermedia Techniques	2-20
2.5.3 Adaptive Sequencing and Adaptive Hypermedia	2-20
2.6 Summary	2-22
CHAPTER 3 : PROJECT DESIGN	3-1
3.1 Introduction	3-1
3.2 Drawbacks of CALI-I.....	3-1
3.3 Platform for CALI-II.....	3-2

3.4 Experimental Setup	3-3
3.5 Structure of CALI-II.....	3-7
3.5.1 Expert Module	3-8
3.5.2 Student Module.....	3-11
3.5.3 Curriculum Module	3-12
3.5.4 Interface Module.....	3-15
3.6 Expected Dialogue	3-17
3.7 Development Platform and Tool	3-21
3.7.1 Programming Languages.....	3-21
3.7.2 Flash.....	3-22
3.7.3 RealSystem	3-24
3.8 Summary	3-25
CHAPTER 4 : DEVELOPMENT OF INTERFACE MODULE	4-1
4.1 Introduction.....	4-1
4.2 Overall Structure and Presenting Methods.....	4-2
4.3 Creating and Streaming Videos.....	4-5
4.4 Creating Animation.....	4-8
4.4.1 Flash Movies.....	4-8
4.4.2 Publishing Flash Movies.....	4-12
4.5 Interactive Playground for Circuit Simulation	4-13
4.5.1 Control of Flash Movies Using Scripting Methods	4-13
4.5.2 Component Selection.....	4-14
4.5.3 The Simulation for Circuit Connection	4-18

4.6 Creating the Pop-up Windows	4-22
4.7 Summary	4-24
CHAPTER 5 : STUDENT MONITOR & GUIDE	5-1
5.1 Introduction	5-1
5.2 Student Monitor	5-1
5.3 Feedback Provided by the Tutor Module	5-9
5.4 Summary	5-17
CHAPTER 6 : PROJECT OUTCOMES AND EVALUATION.....	6-1
6.1 Introduction	6-1
6.2 Login Page and Navigation Frame.....	6-1
6.3 Laboratory Equipment.....	6-4
6.3.1 Graphical Menu	6-4
6.3.2 Component Pop-up Window	6-6
6.4 Project Procedure	6-8
6.4.1 Graphical Menu	6-8
6.4.2 The Pop-Up Window for “What you should know”.....	6-10
6.4.3 The Pop-Up Window for “Procedures”	6-11
6.4.4 Circuit Simulation.....	6-13
6.5 Other Sections	6-17
6.5.1 Project Description, Overview and Assessment	6-17
6.5.2 Facilities: Note Adding, File Uploading, Search Engine and Forum	6-19

6.6 Evaluation	6-19
6.7 Summary	6-28
CHAPTER 7 : CONCLUSION	7-1
7.1 Introduction.....	7-1
7.2 Specific Outcomes.....	7-1
7.3 Generic Outcomes	7-3
7.4 Further Improvement.....	7-4
APPENDIX A : INSTALLATION OF CALI-II.....	A-1
APPENDIX B : TABLE OF FIGURES, TABLES AND LISTS.....	B-1
B.1 Figures	B-1
B.2 Tables.....	B-5
B.3 Lists	B-5
REFERENCES	R-1

Chapter 1 : Introduction

1.1 Project Background

Instruction is a collection of processes including communicating information to the learner, stimulating relevant learning activities, evaluating the results of those activities and taking remedial actions if necessary. Superior instruction has always been individualised. An increase in the number of students has historically changed the nature of instruction to large group-centred or mass-instruction, particularly at universities.

In many engineering courses students are taught the principles of analysis and design of a system through both theory and laboratory work. Through experimental work students develop a better understanding of the theory by applying it to a real system. They also appreciate more the practical implications of the theory.

Individualised instruction becomes more crucial in a practical laboratory. The students require constant attention to ensure that correct procedures are followed and instruments are employed safely. In addition the validity of the results obtained should be also continuously monitored to guarantee that the experimental rig is set up technically correctly and appropriate methodologies are applied. This in practice does not occur as the human resources required for such operation is beyond the capability of the majority of teaching institutions.

In the School of Electrical, Computer and Telecommunications Engineering, University of Wollongong, Control Systems is a core subject taught over two semesters to all the students enrolled in the three BE degrees of Electrical

Engineering, Computer Engineering, and Telecommunication Engineering. It has an annual enrolment of 100 students. The subject has both theoretical and practical components which are closely integrated in their objectives and contents.

The laboratory section of the course consists of several analysis and design experiments based on a DC servo motor with tacho feedback. The design can be carried out both in analogue and digital domains. An input/output card provides data exchange between the rig and the control computer. In order to overcome the problems referred to above, a computer assisted laboratory instruction (CALI) system has been under development over the last three years.

The first version of the package CALI-I was developed two years ago as a stand-alone software running on a desktop PC. A number of shortcomings were identified in the approach and also in the package itself. Hence, it was decided to develop a second version of the package, CALI-II as a Web-based delivery system.

The work reported in this thesis describes the design philosophy, instructional design and implementation of CALI-II.

1.2 Shortcomings of Conventional Laboratory Instruction

In conventional laboratory instruction, the laboratory procedures are described in the laboratory notes which are handed out to students in print form. During the laboratory session, one-to-many or one-to-one on-the-job tutoring is carried out by the laboratory supervisor, usually when students require help. This approach suffers from the following shortcomings:

1. The laboratory notes are often not quite clear on the practical aspects of the experiments such as how a device should be employed or what experimental procedures are to be followed. This is the result of wrong assumption made by the instructor on the level of familiarity of the students with the laboratory and its procedures. This inconsistency has already been the main problem in traditional instruction (Shlechter, 1991).
2. Students do not have the necessary skills to carry out the experiment. Hence they may cause accidents by connecting the wrong circuits or pressing the wrong buttons. Even though a wrong connection may not result in accidents, it will cause wrong measurements and the incorrect results. Students are not usually aware of such mistakes and the instructor almost has no time to actively check each student's practical work and to identify such mistakes.
3. Students usually require tutoring on-demand, particularly when they encounter a problem. This, however, is not possible due to the lack of sufficient human resources. Hence, students often have to wait long periods of time to get the necessary attention from the laboratory instructor.
4. On-the-job tutoring is poorly structured and defined, depending on the problems met by students in real time and the knowledge of the laboratory instructor. Thus, students are not provided with a stable knowledge-base and a complete introduction to the system. In addition, the questions asked by different students are the same and hence it can cause fatigue in the instructor.
5. Students' mistakes are usually identified when the laboratory reports are marked and assessed by the instructor. This is usually too late as no remedial action can be practically taken to fix the problem. Students are also not interested in reviewing their laboratory work in view of the mistakes identified. Thus the

laboratory loses its educational impact. In addition the laboratory reports, often written sometimes after the experiment, may not be the student's own work. Hence, the mark may not reflect the student's ability in the subject (Naghdy and Dedhia, 1997).

1.3 Project Motivation and Objectives

This project had the primary motivation of overcoming the shortcomings of the conventional laboratory instruction by designing a Web-based tutoring system to offer a protected and guided environment in which students can master background knowledge, practise some operational skills, and learn troubleshooting methods.

The **general** goal of this project was threefold:

1. To conduct a review of computer-assisted instruction, computer-assisted laboratory instruction and Intelligent Tutoring Systems to identify the advantages and disadvantages of introducing intelligence into computer-based tutoring.
2. To develop an effective multimedia Intelligent Tutoring System (ITS) as an alternative or complementary approach to conventional laboratory instruction.
3. To evaluate the effectiveness of the developed system by introducing it as part of a main stream subject in the School.

The **primary** goal of developing such a system was to provide students with sufficient knowledge and expertise to enable them to carry out the experiments successfully and effectively. The following objectives were pursued towards the primary goal:

1. To provide well structured and defined background knowledge about the equipment used in the laboratory and the experiment prior to conducting the practical work.
2. To facilitate interaction of the students with the system at their own pace and as often as they need in order to master the necessary background materials.
3. To make the CALI continuously available on the Web independent from the timetabled sessions.
4. To provide a stable and standardised domain knowledge independent from the knowledge-base of the instructor.
5. To provide practical skills through computer simulation to prevent accidents or damage to the equipment.
6. To provide immediate on-demand tutoring to solve the problems encountered by the students.
7. To reduce the load of the laboratory instructor and to relieve him/her for more meaningful tasks.

The **further** aim of the system was to incorporate intelligence and multimedia into the ITS. The related objectives were:

1. To constantly monitor the students' activities and update the Student Model.
2. To provide appropriate and well defined tutoring curriculum and pedagogical strategies.
3. To ensure that the students receive the necessary information when they progress by actively guiding them.
4. To reduce the gap between the student and the domain expert by dynamically adjusting the presented materials and feedbacks.
5. To motivate the students' interest to carry out the experiments effectively.

6. To deliver the instruction through an advanced and pleasant multimedia interface, incorporating graphics, animations, videos, and explanatory text.

Due to difficulty, effort, time and cost associated with the implementation of a true ITS, emulating human-like instruction has never been achieved, even for the most limited domain. Therefore it was appropriate for this project to make a compromise between intelligence and reality to develop a practically achievable ITS.

1.4 Advantages of CALI

1. CALI can provide students with consistent and standardised tutoring on demand and give them immediate feedback. It can include a large database to provide detailed and structured background knowledge on experiments. The knowledge base will not be dependent on the ability and availability of the laboratory supervisor. Every student can receive the same quality tutoring and access the same knowledge-base at any time without waiting for the scheduled laboratory sessions.
2. CALI can reduce the student-to-tutor ratio to one and offer each student the chance to progress at a pace suited to his/her needs or learning capabilities (Dean, 1983). The students can go through some materials more rapidly, or review a particular topic as many times as they need.
3. By incorporating a wide range of media formats, CALI can deliver information in a style and format not possible in printed form. Multimedia, such as graphics, animation, and video, can be used in CALI where necessary to draw students' attention, motivate their interests and provide them with practical information and examples that supplement the theoretical information. It helps to offer

accurate illustration of the theoretical fundamentals, quick delivery of relevant information, and a comprehensive overview of the experiments. For example, many small contextual details in video can help the students to become familiar with both the simulated microworld and the real world.

1.5 ITS Versus CALI

Although CAI and CALI allow students to explore the information at their own pace, they provide information in a pre-determined and rigid form without taking into account the student's background knowledge and learning skills. They can provide rich feedbacks, media and simulations while they cannot adapt to the student's current knowledge state. In such an exploratory learning environment, the page accessed and the knowledge acquired by the student depend on the student's navigational activities that are unguided. Hence, a student may miss to learn important concepts, system components or operational procedures.

To overcome such shortcomings of the traditional CAI system, a more advanced adaptive instructional system, or the Intelligent Tutoring System (ITS) has been developed. ITS adapts the presentation of knowledge and the means of instruction to a particular student by using Artificial Intelligence (AI) techniques.

CAI evolves to ITS by incorporating knowledge representation and student modelling. Knowledge representation includes storage, modification, and manipulation of information. It makes the domain knowledge known to the computer system well enough for the embedded expert to draw inferences or solve problems in the domain, and provides pedagogical knowledge that can apply the best means of instruction to reduce the difference between expert and student performance (Burns

and Capps, 1988). Student modelling enables the system to form a dynamic model for a student, to obtain relevant learning information from a student, and to present relevant materials to optimize the student's learning process. The student model may be influenced by the student's activities, responses and inputs. Intelligent Tutoring Systems use the knowledge about the domain, the student, and teaching strategies to support effective, flexible and individualised learning or tutoring.

Both CAI and ITS are two-way knowledge communication systems which assist teaching of a concept or skill but there is an important difference between traditional CAI systems and Intelligent Tutoring Systems. Traditional CAI does not adapt to the individual student while the ITS can present information by taking into account numerous factors about the individual student. It can react to the student's action and make instructional decisions for each individual according to the context. Therefore, ITS is more flexible.

World-Wide Web (WWW) will not only provide easier access but also rich interactivity and illustrations, incorporating graphics, sound, video and hyperlink. It is a self-sufficient and complete hypermedia system which has been widely used for distance learning. Implementing ITS with WWW is an efficient way to integrate the flexibility and intelligence of ITS with the wide availability and rich interactivity of WWW.

1.6 Structure of this Thesis

Chapter 1, *Introduction*, outlines the problems associated with conventional laboratory instruction, the limitation of CAI, the advantages of ITS over CAI, and the objectives of the project.

Chapter 2, *Background*, presents a literature survey and previous work in the field of CAI, CALI and Intelligent Tutoring Systems. At first, this chapter introduces the concepts of CAI and CALI, and shows their advantages over traditional classroom lecture and laboratory instruction. Then the limitations of conventional CAI, the need for implementing intelligence in CAI, the evolution from CAI to ITS and the characteristics of an Intelligent Tutoring System are explained. The structure and components of a typical ITS are also described. In the end of the chapter, it is pointed out that although the theoretical concept behind intelligent tutoring systems sounds very attractive, these systems are very difficult to implement. Hence, very few ITS systems have successfully developed all the necessary components.

Chapter 3, *Project Design*, discusses the design of the web-based system — CALI-II. Justifications and feasibility of developing such a system are provided, and project requirements are identified. The programming tools selected for the development of CALI-II are introduced. The design of the overall system structure and each module are presented.

Chapter 4, *Development of Interface Module*, describes the implementation of the Interface Module which is the main channel of communication between students and the system. Many issues associated with the development of the interface are examined, such as how instructions can be presented, video streaming can take place and animations and simulations are created.

Chapter 5, *Student Monitor & Guide*, explains the creation of Student Monitor and Guide in this package with emphasis on the building of intelligence. The operation of Student Monitor and Guide are described and the method of their implementation is explained.

Chapter 6, *Project Outcomes*, describes the project outcomes and provides the evaluation results. The evaluation focuses on testing the system and evaluating its effectiveness using a group of students.

Chapter 7, *Conclusion*, summarises the characteristics of the current system, draws some specific and generic conclusions, and proposes further work to make the system more effective and more adaptive to an individual student or a particular group.

Chapter 2 : Background

2.1 Introduction

In this chapter, two concepts of Computer Assisted Instruction (CAI) and Intelligent Tutoring System (ITS) will be studied. Both methods are two-way knowledge communication systems that can provide instructions to the user and assess the user's knowledge, but using quite different approaches.

Traditional instruction is delivered by an educator, who designs the course content, the instruction style and the evaluation method. Computer Assisted Instruction (CAI), on the other hand, uses a computer-based multimedia approach to deliver instruction and assess the students.

With the introduction of second-generation computers in the late 1950's, educationalists responded quickly to the perceived potential for CAI. The PLATO project at University of Illinois, which began in 1960, was one such example (Alpert and Bitzer, 1970).

Now computer assisted instruction (CAI) has been widely accepted in the academic field. It has been used both as an adjunct to traditional instruction and as an alternative instructional method (Schmidt et al., 1991; Wong, 1990; Kohl & Su, 1995; French, 1986). By controlling a variety of media modalities, CAI can enhance instruction and manage the learning environment (Saba & McCormick, 1996). It can also be used for the delivery of laboratory instructions. In this case it is called Computer Assisted Laboratory Instruction (CALI), which presents numerous advantages over traditional laboratory instruction.

Intelligent Tutoring Systems (ITSs), almost a synonym to Intelligent Computer Aided Instruction (ICAI) Systems in most cases, are usually referred to as the resultant combined AI-based systems (Sleeman and Brown 1982). They have improved traditional CAI systems by incorporating Artificial Intelligence techniques into computer-based software packages to assist teaching of a subject or skill. In some cases, ITSs may be considered as the second generation of CAI systems.

Artificial Intelligence (AI) is the study of how to make machines, especially computer systems, do things that require intelligence as done by people. The term “intelligent”, although misleading sometimes, refers to the system’s ability to understand, learn, reason and solve problems. ITSs should have the capacity to know who and what are taught, as well as when and how the teaching should be carried out. In order to perform tasks that seem to depend on human intelligence, ITSs have drawn skills not only from artificial intelligence but also from many other fields such as psychology, education and learning, cognitive science, and human factors.

2.2 Computer Assisted Instruction (CAI)

2.2.1 Terms Related to CAI

There is usually some confusion about the difference between computer-assisted instruction and other methodologies such as computer-aided instruction, computer-based instruction, computer-enriched instruction and computer managed instruction. The word “instruction” in these phrases is sometimes replaced by “learning”, “education” or “training”. However, according to Bangert-Drowns, et al. (1985), Batey (1987), Grimes (1977) and (Marshall 1988), there are some differences among these terminologies:

- Computer-based instruction (CBI) is the broadest term given to almost any kind of computer application in educational environments.
- Computer-managed instruction (CMI) mainly utilises computers to organize student data, evaluate students' test performance, guide them to appropriate instructional resources, and make instructional decisions.
- Computer-enriched instruction (CEI) is a learning activity in which computers execute programs developed by the students, or enrich relatively unstructured exercises designed to motivate students.
- Computer-assisted instruction (CAI) most often refers to drill & practice, tutorial, simulation, problem solving or instruction games.
- Computer-Based Training (CBT) refers to software applications in which the computer acts as a human tutor in education or training.
- Computer-aided instruction (CaI) includes computer-aided testing, CMI and classroom management. It doesn't emphasize the learning and teaching functions as computer-assisted instruction.

Such definitions are not universally accepted by everyone and there is some disagreement.

Computer Assisted Laboratory Instruction (CALI) has its main focus on laboratory instruction. CALI can also refer to Computer Assisted Language Instruction or Computer Assisted Law Instruction. In this thesis, however, the reference is mainly to Computer Assisted Laboratory Instruction.

2.2.2 Comparison of CAI and TCL (traditional classroom lecture)

In the literature, the CAI and Traditional Classroom Lecture have been extensively compared in various studies.

Computers can repeat an instruction as many times as needed, for students to master the contents. Thus CAI can reduce drudgery and repetition in teachers' work, free teachers for more productive work with the students and help instructors to focus on more difficult and essential problems (Kulik, Bangert & Williams, 1983).

CAI has potential as an instructional medium to individualize the learning process (Allen, 1986; Neil, 1985). The student can work through the material along his/her own path of learning, at his/her own pace and at a time convenient to him/her.

With the increased active information exchange between computers and students, computer-assisted instruction can enhance the way people think (Salomon et al., 1991).

In addition, CAI employs many different teaching strategies to meet the various learning needs of different students.

- People with left-brain dominance are primarily auditory learners, and those with right-brain dominance are primarily visual learners (Springer & Deutsch, 1981; Vitale, 1981). CAI can provide both sound and graphics to activate left and right brain at the same time.
- According to Khoiny (1995), with the concrete experiences provided by CAI, CAI would greatly benefit accommodators and divergers among the four learning styles (assimilators, accommodators, divergers, and convergers) classified by Kolb (1976).
- Fletcher (1990) pointed out that people retain only 20% of what they hear; 40% of what they see and hear; and 75% of what they see, hear and do. The interactive nature of CAI can integrate the three senses (sight, hearing and touch) to be used in the learning process.

CAI can provide immediate and automatic feedbacks. In CAI students' correct answers to posed questions can be rewarded immediately. Thus a CAI program will outperform a frame of text due to the fact that if the occurrence of an operant is followed by the presentation of a reinforcing stimulus, the strength is increased (Skinner, 1958).

Some research has shown that the application of CAI as a supplement to traditional teacher-directed instruction, enhances the learning processes and assists students to achieve more than when they are exposed to traditional instruction alone (Bangert-Drowns, et al. 1985; Batey 1986; Braun 1990; Capper and Copple 1985; Edwards, et al. 1975; Grimes 1977; Kann 1987; Kulik, Kulik, and Bangert-Drowns 1985; Martin 1973; Rapaport and Savard 1980; Rupe 1986)

The research has also shown that students learn faster when exposed to CAI than when taught by conventional instruction. According to Capper and Copple (1985) the learning rate is sometimes 40 percent faster. (Batey 1986; Capper and Copple 1985; Edwards, et al. 1975; Grimes 1977; Hasselbring 1984; Kulik 1983, 1985; Kulik, Bangert, and Williams 1983; Kulik and Kulik 1987; Rapaport and Savard 1980; Rupe 1986; Stennett 1985; White 1983.)

The retention of the content learned through CAI has also proved to be superior compared to retention with TCL alone. (Capper and Copple 1985; Grimes 1977; Kulik 1985; Kulik, Bangert, and Williams 1983; Kulik, Kulik, and Bangert-Drowns 1985; Rupe 1986; Stennett 1985; Woodward, Carnine, and Gersten 1988.)

2.2.3 Web Based CAI

Computer assisted instruction is usually offered in a variety of software packages on microcomputers (Bolwell, 1988). With the proliferation of the World Wide Web

(WWW) into the educational field (Nichols, 1995; Whittington, 1996). a large number of distance education programs have emerged (e.g., Howard. Ault. Knowlton, & Swall, 1992; Johnson & Amundsen, 1983) as the Internet allows anyone from anywhere to participate in a course offered on the Internet (Treuhaft. 1995). Moreover, web based CAI provides much more interactivity and illustrations. incorporating graphics, sound, video and hyperlink, to motivate students' interest in studying the materials (Khoiny, 1995; Gibson et. al). Hence, more and more CAI programs are based on the World Wide Web (Stauffer, 1996) for easy access and effective interface.

2.2.4 Evaluating a CAI System

According to Bolwell (1988), Parks (1996) and French (1986), an effective CAI should provide the following features:

- ease of use
- clear instructions and objectives
- allowing active participation and individual differences
- providing a variety of accurate and specific feedbacks
- positive reinforcement or repetition of concepts
- maximizing interaction and interactive capabilities
- maintaining learners' interests
- using the computer's resources wisely

In addition, a web based CAI program should have a simple navigation, should be fast to download, and include effective hyperlinks to connect to appropriate and applicable resources (University of Maine, 1995).

2.2.5 Computer Assisted Laboratory Instruction (CALI)

In a practical laboratory, accidents may happen due to unawareness of the students of the procedures and experimental details. Therefore, it is essential that students understand equipment and are familiar with the proper operational procedures in order to prevent the occurrence of accidents caused by wrong connection of circuitry or pressing the wrong buttons, etc.. CALI can draw the students' attention to proper procedures, and motivate their interests in mastering the theoretical fundamentals.

Students usually have to wait to ask questions from the laboratory supervisor (Tira, 1998), and they usually have to wait until their questions are solved to continue their experiments. Many questions asked by different students are also similar. CALI can reduce the instructors' burden and help them to focus on more important problems.

Some students may use a wrong circuit or measurement procedure during the entire experiment, resulting in incorrect results. Due to the number of students involved, instructors usually cannot identify such mistakes until the students' reports are marked. Students become aware of their mistakes when they receive their reports and have obtained their marks, which is rather too late. Properly designed CALI (Araki, et al, 1993) can monitor students' progress, advise students of their weak points and notify these to teachers. Thus students can get immediate and effective feedback.

CALI can record and evaluate students' progress by including CMI (computer managed instruction) functions. This provides more facts to reflect the student's ability in addition to the laboratory reports (which may not be the student's own work).

A number of research groups have made attempts to develop CALI systems to overcome some of the problems mentioned above.

A cost effective interactive multimedia system for electrical undergraduate laboratory sessions has been developed by Chan and So (1994). This package is built on Microsoft DOS environment. It includes: engineering circuit analysis; characteristics of induction machines; characteristics of d.c. machines; motor speed control by static converter and inverter; etc.

Another system titled Computer Assisted Instruction learning system has been developed by Araki, et al (1993). This system consists of three parts: the pre-Lab-CAI; the laboratory; and the post-Lab-CAI, with the pre-and post-Lab-CAI made up of quiz-type CAI and tutorial type CAI.

In another work, an integrated laboratory environment has been developed by Kozick and Crane (1996) for system modeling, simulation, real-time digital signal processing, and control. This project uses the visual programming environment of Simulink (registered trademarks of the MathWorks, Inc.) to provide a common interface for both computer simulation and real-time execution of algorithms on the DSP (Digital Signal Processing) hardware, with the simulation software running on workstations and all the laboratory equipment (including the DSP hardware) connected to the Internet. Students repeat the process of modeling, simulation and experimental testing until computer simulation results agree with the experimental measurements of the actual systems.

There are also two on-line CALI programs available on the Web:

- Instrumentation and Optics Laboratories (maintained by Arietas, on-line)
- Introductory Laboratories in Biology (maintained by Steinwand, on-line)

2.3 Evolution from CAI to ITS (Intelligent Tutoring System)

Computer Assisted Instruction (CAI), started in the 1950s, was influenced by learning machines and operant psychology (Pressey, 1927; Skinner, 1958). Skinner's behaviourist psychological theories, which suggest that the presentation of a reinforcing stimulus following the occurrence of an operant will increase the strength, provided the basis for the linear programmed-instruction methodology. With simple "linear programs", CAI allows a student to work through the material at his/her own pace and rewards his/her correct replies immediately.

Then "branching programs" were used in the 1960s to control the next material according to the student's response, i.e., to adapt the teaching to students' performance. In the 1970s "generative" systems were discovered in the design of CAI systems in some domains such as arithmetic, where the teaching materials can be generated by the computer itself from the combinations of different elements in a large database (Uttal, Rogers, Hieronymous, & Pasich, 1969; Koffman & Blout, 1975). Also, the "separated" structure for CAI was developed (Stelzer & Garneau, 1972), in which subject content and instructional decision rules were separated in different files. This structure is convenient for system maintenance and modification. CAI systems have improved from the programmed learning machines to advanced interactive teaching devices with the richness of feedback and increased degree of individualisation. However, they are still far short of human-like knowledge and ability. In CAI, the instruction is delivered in a rigid manner that is set by the designer. For all users, it is almost the same.

Recognising the deficiencies of traditional CAI systems, Intelligent Tutoring Systems (ITS) were subsequently developed to adapt the presentation of knowledge and the

means of instruction to a particular student. As Intelligent CAI of the 1980s, Intelligent Tutoring Systems (ITS) started as an effort to overcome shortcomings of generative systems and to develop more powerful adaptive instructional system by using Artificial Intelligence (AI) techniques, which have dealt with how best to represent knowledge. Then ITSs were used in many areas such as Language Learning (Imlah and du Boualey 1985; Barchan, Woodmansee and Yazdani 1985), Electronic Trouble Shooting (Brown, Burton and de Kleer 1982) and Computer Programming (Anderson and Reiser, 1985; Johnson and Soloway, 1985).

Early ITSs incorporated an expert system. An expert system uses inference to apply knowledge to provide expert-like solutions to problems in a specific domain (Alvey 1982; Bratko 1991). An expert system can describe its decisions and the underlying reasoning. A simple expert system architecture is shown in Figure 2-1.

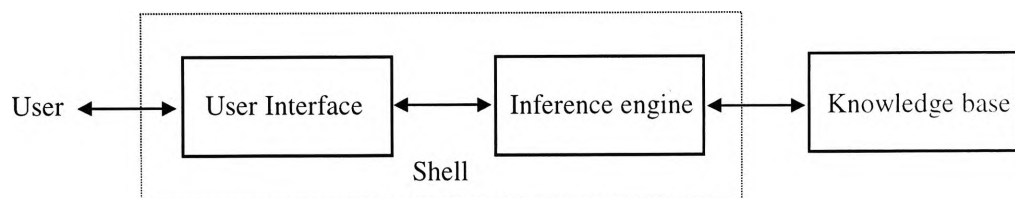


Figure 2-1: A Simple Architecture of an Expert System

Expert systems can be used not only for automating some part of human decision making or reasoning but also for imparting knowledge from an expert in a field to a large number of trainees (O'Shea and Self 1983). A number of ITSs have been built around pre-existing expert systems. GUIDON (Clancey 1979) is such an example which is produced by implementing an expert system called MYCIN (Shortliffe 1976) as the expert model. GUIDON was not a very successful instructional system, which "suggested that an expert system's knowledge base is not appropriate to use as

the knowledge base of an ITS system unless supplemented by other levels of the knowledge that help explain and organize the knowledge in the teaching process” (Kearsley, 1987).

Although most ITS systems are based on simulating the student’s behaviour, they use different methods and structures to realise students’ mistakes and to offer appropriate advice (Sleeman and Brown 1982). Some examples of ITS structures are introduced below (Yazdani, on-line).

1. Adaptive Control of Thought (ACT) ITS architecture consists of domain expert (or “ideal student” model capable of solving problems in the domain), bug catalogue (or an extensive library of common misconceptions and errors in a domain), tutoring knowledge (containing the strategies used to teach the domain knowledge), and user interface. (Anderson 1982; Anderson and Reiser, 1985; Anderson, Boyle and Yost, 1985; Anderson, Boyle and Reiser 1985).
2. Hartley and Sleeman (1973) suggested that an ITS should normally have four distinct knowledge bases: knowledge of the task domain, a model/history of the student’s behaviour, a list of possible teaching operations, and mean-ends guidance rules which relates teaching decisions to conditions in the student mode.
3. The five-ring model presented by O’Shea et al (1984) includes student history, student model, teaching strategy, teaching generator, and teaching administrator.

O’Shea et al’s (1984) proposal is closer to the traditional CAI systems which emphasises general purpose teaching skills while Anderson's proposal is similar to the more open-ended problem solving environments of Papert (1980) and Yazdani

(1984) which care about the representation of the knowledge domain. Hartley and Sleeman's (1973) proposal is in between. The problem-solving (or exploratory learning) environment is more suitable for teaching exceptionally abstract and general concepts by constructing a proper computer-based microworld (Lawler, 1984).

A typical ITS separates the major components of an instruction system into the contents to be taught, the instructional strategy, and a mechanism for identifying what the student has understood or possibly misunderstood. In an ITS, these three components are identified and referred to as the expert, the curriculum and the student module respectively (Burns and Capps 1988; Anderson 1988). The communication process of an ITS with the students takes place through the user interface. Hence a well-established architecture for a complete Intelligent Tutoring System consists of four interconnected software modules: the expert module, the student module, the curriculum and diagnosis module (or the tutor module), and the environment module (or the user interface module).

In conclusion, CAI evolves to ITS by making the domain knowledge known to the computer system well enough for the embedded expert to draw inferences or solve problems in the domain, by enabling the system to obtain relevant learning information from a student, and by establishing an intelligent tutorial or pedagogical mechanism that can apply the best means of instruction to reduce the difference between expert and student performance (Burns and Capps, 1988). Therefore, ITS is more flexible. It can react to the student's action and make instructional decisions needed for each individual user according to the context. A comparison of CAI and ITS is shown in Table 2-1. The main differences between CAI and ITS lie in the knowledge structure, the presentation of knowledge and the modeling of the student.

Table 2-1 A Comparison of CAI and ITS (Kearsley, 1987)

Issue	CAI	ITS
Development Goals	Tradition CAI has been developed by educational researchers and training developers to provide instructionally effective and efficient systems by applying computer technology.	ITS has been initiated primarily by computer scientists to explore AI techniques in instruction.
Theoretical bases	Although not developed on a strong theoretical base, most CAI systems have incorporated learning theories and instructional principles in one form or another. For example, the early forms of CAI were influenced by Skinnerian behaviourism.	Many ITSs are designed and built on theoretical notions of cognitive science that have grown out of information-processing theory in cognitive psychology.
Structure and process of the system	<i>Frame-oriented static structure; predefined, system-initiative process</i> In most CAI systems, all of the instructional components are stored and implemented in a single structure. Although some systems have separate modules, their operational procedures are still determined by predefined processes.	<i>Process-oriented dynamic structures; generative, mixed-initiative processes</i> ITSs can use spontaneous inferential processes to diagnose the student's learning needs and prescribe instructional treatments.
Instructional methods	<i>Various instructional methods, including expository, discovery, and combined approaches</i> Most CAI methods take a teacher-centered expository form, which requires the student first to understand the teacher's instruction and then to practice given questions to reinforce this understanding.	<i>Mainly discovery approaches</i> The basic instructional method for ITSs is "learning-by-doing" (Dewey, 1910; Sleeman & Brown, 1982) with its conditions referred to as the "reactive learning environment" (Brown, Burton, & de Kleer, 1982). Thus in most ITSs, instructional methods take a student-centered expository form, and tutorial dialogues are basically determined by the student's conceptual understandings and learning behaviours.
Methods of structuring knowledge	<i>Mainly task analysis for identifying subtasks and content elements</i> Methods of task analysis include an algorithmic approach, a hierarchical approach and their combination.	<i>AI knowledge representation techniques for organizing knowledge into a data structure</i>
Student modeling	<i>Binary judgment of student responses (eg, correct or incorrect); prespecified response-sensitive procedures; quantitative method</i> In a quantitative model, students' learning is characterized in probabilistic terms.	<i>Qualitative evaluation of student responses</i> In a qualitative model, students' learning is assessed from the analysis of their response (or response patterns). The modelling is a process of making inferences about their conceptions and misconceptions.

2.4 Intelligent Tutoring Systems

2.4.1 Structural Model of ITS

A typical ITS consists of four modules: the expert module, the student module, the tutor module, and the interface module. Figure 2.2 represents the typical structure of an ITS.

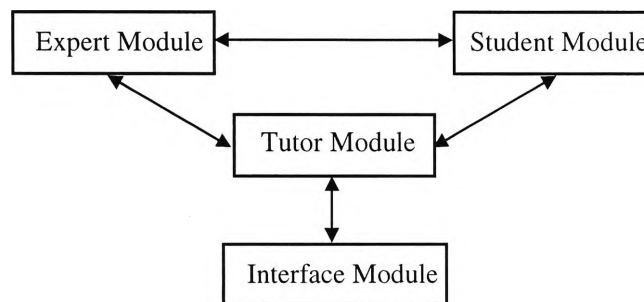


Figure 2-2 Typical Structure of an ITS

In the following sub-sections, each module is described.

2.4.2 The Expert Module

The Expert Module contains the objects of the instructions such as facts, concepts, processes and skills. This module interfaces with the domain knowledge embedded in the system. It provides a dynamic form of domain expertise by manipulating an Expert Model in the knowledge domain.

A number of AI knowledge representation methods have been developed previously. *Semantic networks* has been developed based on psychological models of human associative memory (Norman & Rumelhart, 1975; Quillian, 1968). A semantic network represents the domain knowledge (including objects, concepts, and situations) as a set of nodes linked by labelled arcs that represent the relationships

between the nodes. The method incorporates all the necessary information in a large, static database and makes inferences by following the arcs between the nodes.

Production systems have been developed by Newell and Simon (1972) for their models of human cognition, and have been used as a way of modelling people's behaviour in the domain of tutoring to represent skills and problem-solving methods. A production system organises knowledge into three different categories (Yazdani, on-line):

- Facts: Factual (declarative) knowledge about a particular case.
- Rules: Procedural knowledge on how to reason in a domain and expertise (in the following form: IF *condition(s)* THEN *conclusion(s)*)
- Inference: Control (qualitative) knowledge of how to carry out reasoning from a set of given facts and rules to come up with a conclusion.

The basic idea of production systems is that the If-Then rules (also called production rules or productions) can encode many forms of knowledge (Anderson 1985). Hence, this method is by far the most widely used method of representing domain knowledge in expert systems.

There are also If - Then rules with uncertainty measures to indicate confidence levels associated with a production rule (Rich and Knight 1991).

Script-frames have been developed by Schank and Abelson (1977) to represent event sequences on the basis of Minsky's (1975) work. Script-frames are data structures including declarative and procedural knowledge in prespecified internal relations. A frame is a collection of attributes, values and procedures. Frames can be used to simplify semantic networks. They can also be used in production systems with frames defining the objects occurring in the rules.

There are also other types of representative AI methods such as logic, analogical representations and procedural representations (Wingrad, 1975; Barr & Feigenbaum, 1981).

2.4.3 The Student Module

The Student Module contains information about the student's understanding of the knowledge domain and provides a dynamic vision of the student's knowledge of the domain by manipulating a Student Model that stores the student's current state about the subject. It uses diagnostic tools contained within the Curriculum Module to identify the understanding level of a student of the domain knowledge. It is hard to find a perfect accurate model of the student and even a perfect model will not insure making reasonable pedagogical decisions.

The relevant knowledge about the student, stored in a student model, can be used to make the learning environment adaptable to individual learners' needs (McCalla and Greer 1991). Student models may record either misconceptions, missing conceptions or a combination of both. Missing conceptions are some knowledge which are possessed by the domain expert but not by the student while a misconception is knowledge that is possessed by the student but not by the expert (VanLehn 1988).

The overlay Student model, named by Carr and Goldstein (1977), represents the student's knowledge state as a subset of an expert's knowledge base and assumes that the student will not learn anything that the expert does not know. With the overlay model, the goal of tutoring is to grow the student's subset of the expert knowledge. Constructed by comparing the student's behaviour to the computer-based expert's performance on the same task, this model does not cater for misconceptions or bugs not presented in the expert domain knowledge.

The Differential Student model (Burton & Brown 1982) does not cater for student misconceptions or bugs either. It extends the overlay model by separating already presented knowledge from the knowledge that the student has not yet been exposed to. This is useful for making the tutoring strategy.

The perturbation or buggy Student model, proposed by Brown and Burton (1978b), caters for misconceptions or bugs by adding a bug library to the expert knowledge. Within this model, domain knowledge is represented as rules while potential misconceptions as “mal-rules” (Sleeman, 1982) and the goal of tutoring is to enlarge the student’s subset of the experts knowledge while eliminating any misconception.

2.4.4 The Curriculum and Diagnosis Module

The Curriculum Module, also called tutoring module, contains instructions and rules, teaching strategies or other decision-making tools to provide pedagogical expertise to reduce the knowledge differences between the expert and the student. It feeds proper instructions or remediations to the Interface Module at proper point by using rules to judge how well student response (as represented by the Student Model) matches domain knowledge (as represented by the Expert Model). Representing the communication skill, this module has the potential to adapt to the particular learner and to improve learning strategy over time.

This module can give a student different degrees of control over the learning activities. While only monitoring the student’s activity and adapting system responses to student actions, the module does not provide any control. On the other hand, it can provide full control by using guided-discovery learning or coached activities. Control can also be shared between the student and the system when questions and answers are exchanged between them.

A number of methods have been developed for instructional strategies. The *coaching method* provides an environment in which students can learn general problem-solving skills and related skills such as operational skills. The tutor module acts as a coach and tries to maintain student's interest (Burton & Brown, 1982; Goldstein, 1982). Computer games may be used to enhance the learning of the students.

The *Socratic method* provides students with questions to let them debug and modify their own misconceptions (Carbonell, 1970; Stevens, Collins, & Goldin, 1979).

2.4.5 The Interface Module

The Interface Module, as a mechanism for communications between the ITS and the student, presents the user with a uniform environment to facilitate learning. It cooperates with the student and pedagogical modules. Instruction, diagnosis, remediation, and user driven learning may occur in the environment made up of situations, activities, and tools. An effective Interface Module should have the following features:

- clarity of presentation
- ease of use
- attraction in design
- offering obvious interaction and control facilities
- enriching the communication bandwidth using media technology,

2.4.6 Some Remarks on ITS

The theoretical concept behind intelligent tutoring systems sounds very attractive. In practice, however, ITSs have proved to be far from "intelligent". Computers can only

react as they are programmed to. They never think or react like humans. Intelligent tutoring systems are less mature while more costly than CAI.

Some ITSs can be easily changed from application in one domain to another similar domain (Davies et al 1985). More and more attempts are made to develop author languages toolkits for ITS.

In addition, due to the size and complexity of most ITSs, not all of the “Traditional Trinity” of ITSs (the expert, the curriculum and the student module) are fully developed in every system (Clancey 1979). Most useable systems focus on the development of only a single component.

2.5 Contributing Techniques and Technologies

The advances made in computer technology, both software and hardware, have created new tools for enhancing the effectiveness of ITS and CAI. Some of those techniques and technologies are reviewed here.

2.5.1 Multimedia

Rapid progress in computer technology and GUI (Graphical User Interface) has made multimedia very popular particularly in instruction. Multimedia is the combination of graphics, sound, animation, and video, which may be accessed independently, or simultaneously in a single computer environment (Miller 1992). Usually, different students prefer different learning styles. Some prefer texts; others graphics and videos; while a third group may prefer animation or interactive playground. Multimedia techniques can incorporate different media in a single package to provide

multimodal learning styles. In addition, exploring information in various media formats can activate different senses and improve learning.

2.5.2 Hypertext and Hypermedia Techniques

In laboratory instruction, it is important to assist students to become familiar with laboratory equipment, operational procedures and fundamental theories. The conventional approach is to provide the necessary guidance in printed form. However, a traditional sequential paper representation is often boring for the students as it cannot adapt to the interests or needs of a particular student. Stored in the computer as a network of nodes linked together by hotspots or hotlinks, a hypertext representation partly overcomes the shortcomings of a paper representation by allowing rapid and unrestricted jumps between information when hotlinks are selected. This non-sequential, non-linear method of representing and accessing information makes hypertext convenient to use, easy to produce, distribute and update.

Hypermedia extends the idea of hypertext by integrating hotlinks with multimedia such as animation, video and sound. A hypermedia system is a complex information management system providing flexible access to richly connected multimedia. It can organise information in different ways, refer to the same multimedia file from different places and provide direct access to a reference. A good hypermedia system presents knowledge in a meaningful way and provides an environment that encourages the learner to browse and hunt, rather than read from beginning to end.

2.5.3 Adaptive Sequencing and Adaptive Hypermedia

If there are a large number of nodes and links in a hypertext or hypermedia system, a

student can easily lose his/her path in the navigation system. A sequencing proposal adapted to the student's current knowledge and interest will prevent the student from losing his path and will help him/her to master the domain knowledge in a shorter learning time.

Adaptive sequencing, or intelligent knowledge sequencing and problem sequencing, is an ITS technique that adaptively chooses or generates the next teaching operation such as the example, test and question according to the student model and the subject knowledge. There are two different approaches (Specht, 1998):

Adaptation of unit sequence

Adaptive unit sequencing generates the unit sequence according to the specified student's interests and his/her current knowledge to keep him/her on an optimal path. When a student selects a unit, the system can determine whether the student lacks any prerequisite knowledge and can provide the next best unit.

Adaptation of teaching strategy

The adaptation to a preferred learning style or teaching strategy takes into account the student's success and interests with the style. For example, the teaching strategy frequently requested by a student may be first giving some fundamental knowledge, then giving some examples and finally giving some tests. If the student shows satisfactory results in the tests, then this strategy is considered successful and is taken as the default strategy for this student.

Adaptive hypermedia is an adaptive sequencing used in a hypermedia system, where dynamic "intelligent button" activates intelligent knowledge sequencing mechanism. Adaptive hypermedia adapts its work to a particular student by intelligent sequencing of hypermedia pages (contents) and hypernodes (links), i.e., adaptive presentation and adaptive navigation support. Adaptive presentation is a

content-level adaptation by which the content of a hypermedia page is generated or assembled from pieces according to the student's knowledge state. Adaptive navigation support is link-level adaptation including the following navigational adaptation techniques:

Adaptive Annotation

The annotation of a hyperlink is an additional message giving a student some brief information about the content behind the hyperlink. Adaptive annotation provides annotations adapted to a particular student taking into account his/her knowledge and interest.

Hiding or incremental (disable) linking

The multitude of links in hypermedia may cause the problem of overload, i.e., too many alternative ways to proceed. As a solution, incremental linking restricts a student's browsing freedom and incrementally introduces all the materials to the student by hiding some of the links and gradually exposing them.

Ordering

This technology adaptively orders visible links according to the difficulty, complexity and the already mastered areas of the materials.

Combining adaptive hypermedia with free browsing bridges the gap between computer-controlled learning modes and student-controlled learning modes.

2.6 Summary

This chapter provided basic concepts of CAI, CALI and ITS and reviewed some of the previous work. Evolution from CAI to ITS and relationship between CAI, CALI and ITS were also discussed. The importance and significance of using CAI, CALI

and ITS for instruction, especially for laboratory instruction were highlighted. Contributing techniques associated with the development of CAI, CALI and ITS were also introduced.

Chapter 3 : Project Design

3.1 Introduction

This chapter primarily deals with the specifications of the Web-based CALI (CALI-II) and the parameters which affect its design. Initially a background on CALI-I will be provided to highlight its drawbacks and shortcomings. The selection of Web as the implementation platform for CALI-II will be then justified. The rest of the chapter will deal with different aspects of the design including the overall structure of the package, the specifications of its different modules, and the development tools chosen for its programming.

3.2 Drawbacks of CALI-I

This work is a second attempt at developing the Computer Assisted Laboratory Instruction (CALI) for the Control Systems laboratory. The first version of this package, CALI-I, was developed in 1998 (Tira, 1998) using an authoring multimedia software called Quest Net+ (Allen Communication: Quest, on-line). Quest is an objected oriented visual programming tool. In spite of its high level functions and instructions, creating a hypermedia-structured system with animations and simulations based on Quest requires a great deal of coding. Quest is not, however, designed for complex large software programs as the speed of the execution decreases as the size of the program grows.

3.3 Platform for CALI-II

The implementation platform was one of the problems encountered in the development of CALI-I. The size of the program grew too long for debugging, and accordingly the execution of the code became significantly slow. In addition, the code required a large amount of memory. Another shortcoming was that the incorporation of long video segments and facilities to monitor student's navigation activities were proved to be too difficult.

In order to overcome such problems, the second version of the package was intended for delivery on the World Wide Web (WWW), which is a self-sufficient and complete hypermedia system, widely used for distance-learning. Hypertext and hypermedia are the main paradigms for content structuring on WWW. Using web-related languages, techniques and authoring tools, creation of small-sized multimedia files and publishing them on the Web has proved to be quite efficient and simple. Other advantages have also been realised by pursuing WWW-based methods. Web browsers have become an integral part of any computer system and the majority of students are familiar with the use of Web browsers. A WWW-based package will be available to all the client browsers when it is implemented on the School server. Using Common Gateway Interface (CGI) and database techniques on Internet, student's browsing activities can be recorded and guided, as the Web browsers are scriptable and recordable. Hence, Intelligent Tutoring techniques such as adaptive sequencing and adaptive hypermedia can be ported into WWW.

Using WWW platform, it will be possible to develop self-sufficient Intelligent Tutoring Systems (ITS), and to integrate the flexibility and intelligence of ITS with

the wide availability and rich interactivity of WWW. Incorporation of graphics, movies, animation, videos and explanatory text into its multimedia interface, will produce an intelligent multimedia tutoring system (IMTS) or a hypermedia-based ITS. An IMTS also uses the knowledge about the domain, the student, and teaching strategies to support effective, flexible and individualised learning or tutoring, while the power of hypermedia is enhanced by monitoring the student's performance and then reasoning about what to present next and how to present it. Adaptive hypermedia techniques will be used to dynamically guide a student to reduce the distance between the student and the expert.

3.4 Experimental Setup

The control laboratory provides facilities to conduct both analogue and digital control. The experiments in analog control include:

- (a) Transient Response Analysis
- (b) Frequency Response Analysis
- (c) Proportional Control
- (d) PI Control
- (e) Lead Compensation

Experiments in digital control have a similar theme but the controller is a discrete-time control system. Experiments are arranged according to their increasing difficulties and the required knowledge. The laboratory exercises assist the students to evaluate different control schemes in terms of their effectiveness in controlling a

DC motor and to obtain the best possible performance from a system by designing an appropriate controller for them.

Students are expected to use the laboratory equipment and connect associated circuitry to measure the system's characteristics and find out the performances of different controllers. Students work in groups of two. On the completion of each milestone, the group demonstrates the outcomes of that milestone to the laboratory supervisor. A report is submitted at the end of all the experiments. The laboratory equipment for the experimental bench is illustrated in Figure 3-1. Apart from standard equipment such as the oscilloscope and signal generators, the other components of the laboratory include:

The components of the MS150 DC System:

- OU150A Operational Amplifier Unit
- AU150B Attenuator Unit
- PA150C Pre-amplifier Unit
- SA150D Servo Amplifier
- PS150E Power Supply
- MT150F Motor Unit
- IP150H Input Potentiometer
- OP150K Output Potentiometer
- LU150L Load Unit (Brake Assembly)

The digital control, requires more components and software packages including:

- DACBoard (Sabitzer, 1998) (Figure 3-2) developed on the base of MICE Board (Cuifo, on-line 1),
- Hitech C compiler (Cuifo, on-line 2)

- A GUI interface Software, called Digilab, connecting the user with the MICE board (Sabitzer, 1998) (Figure 3-3).

The CALI-II has got components for both the analogue and digital control. In this thesis, the focus is on instruction for the analogue control experiment as the digital one is very similar.



Figure 3-1: Laboratory Equipment

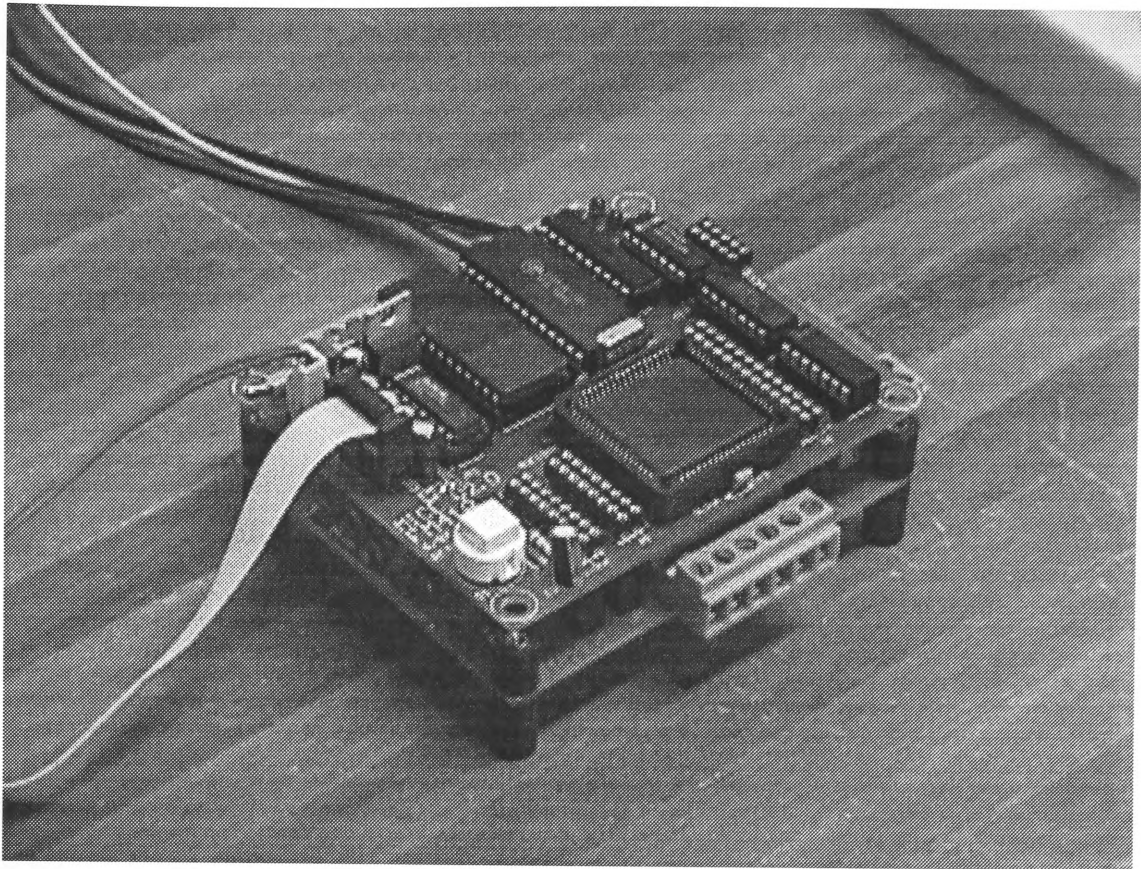


Figure 3-2: DACBoard Used for Digital Motor Control

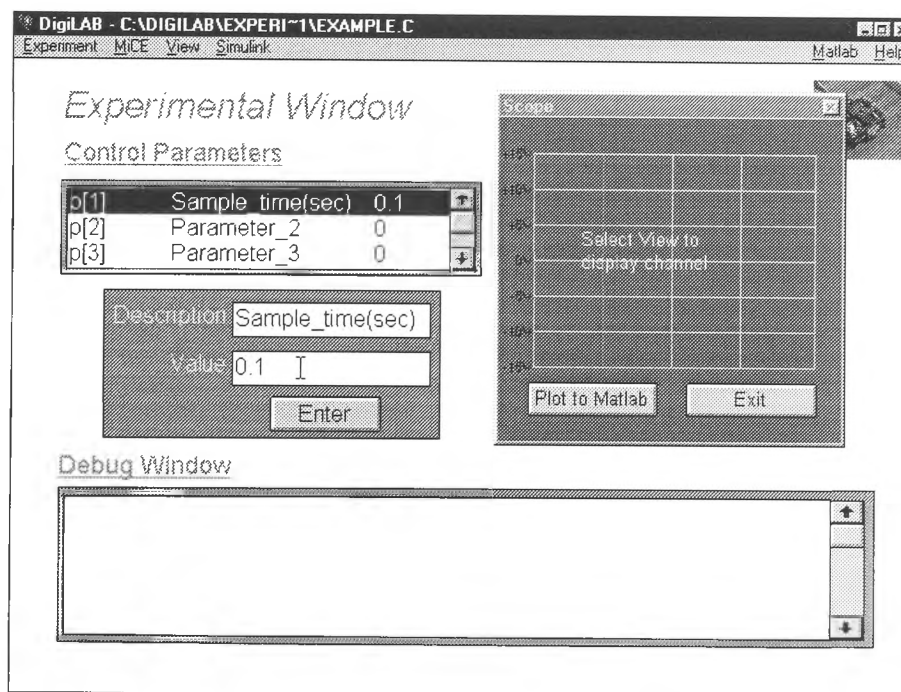


Figure 3-3: "DigiLab" Package Used for Digital Motor Control

3.5 Structure of CALI-II

An ITS typically consists of four modules; *expert module*, *student module*, *tutor* or *curriculum module* and *interface module*. The structure adopted for CALI-II is shown in Figure 3-4.

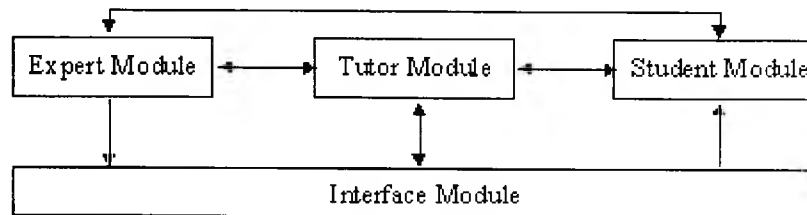


Figure 3-4: Overall Structure

The *Expert module* provides a source for the knowledge (presented by the *interface module*) and serves as a standard for student evaluation (processed by the *tutor module*). In the *student module*, a model of the student's knowledge is generated to enable the *tutor module* to compare the knowledge of the student with that of the expert to evaluate the student's knowledge state and to adjust the instructional strategies. The *student model* is continually updated according to the student's interactions with the *interface module*. The *tutor module*, also called *curriculum module*, contains instructions, teaching strategies, and rules or other decision-making tools to provide pedagogical expertise to reduce the knowledge difference between the expert and the student. It feeds proper instructions or remediations to the *Interface Module* at proper point by using rules to evaluate the student's current knowledge (as represented by the *Student Model*) according to the domain knowledge (as represented by the *Expert Model*). The *user interface* in this project is

handled by the web browser. The design details for each module will be provided in the following sections.

3.5.1 Expert Module

Students are required to do five experiments (as mentioned in 3.4) for analog control (ELEC344). Hence this module should provide sufficient details on each experiment including the technical and theoretical key points emphasised by the experiment, the laboratory component and equipment to be used, and the experimental procedure to be followed.

Thus the instructional contents of the module can be classified into three types of domain knowledge:

- Declarative knowledge: fundamental knowledge or facts about the experiments, objectives, components, overview and assessment
- Procedural knowledge: procedures, simulations
- Operative skill: trouble management, tips for complicated operations

A meaningful and comprehensive structure of the domain knowledge will greatly enhance the presentation of the materials and the learning process.

In order to design the expert module of CALI-II systematically, the *domain model network* (similar to semantic networks introduced in 2.4.2) which has been extensively used in the development of ITSSs has been employed. The domain model network, also called domain network, is a network of nodes and links. The nodes in the network correspond to various elements forming the domain and the links represent the relationships between the nodes. A semantic tree designed for the domain network of CALI-II is illustrated in Figure 3-5.

The subject matters contained in the domain model network are structured hierarchically. The overall content of the domain will be delivered through 5 major nodes of Laboratory Overview, Laboratory Equipment, Project Description, Project Procedures and Report and Assessment. Each node will be in turn broken down into smaller nodes providing more specific information.

For example, Laboratory Equipment node will break down into ten sub-nodes corresponding to ten components used in the laboratory. Each sub-node itself will link to other sub-nodes representing “Tips” on a component, “Some Basic Connections” using the component, and the diagram of a “Typical Circuit” showing the transfer function of the component.

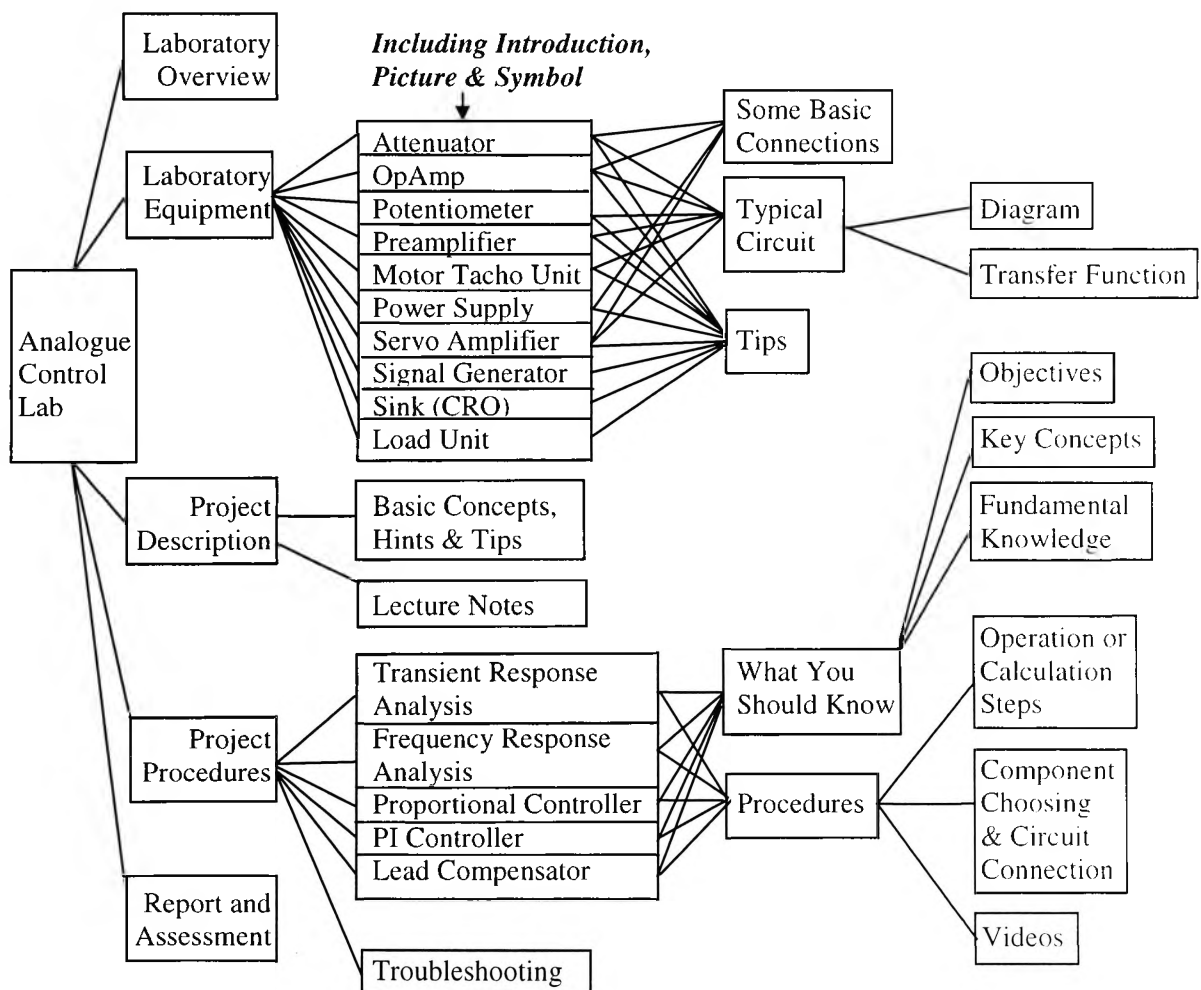


Figure 3-5: Knowledge Domain Network

Different types of learning materials with different pedagogical functionality will be assigned to a node to enhance the contents and the learning processes associated with it. For example, the “Project Procedures” node will link to the following nodes for each experiment:

- *Objectives* which indicate the educational outcomes of the experiment including the knowledge and skills to be acquired and mastered.
- *Key Concepts* which highlight the key concepts used in the experiment so that students can strengthen their understanding on those concepts if they have some shortcomings.
- *Experiment Description* (including fundamental knowledge and operational steps) which provides detailed information on the theoretical and procedural aspects of the experiment through text, graphics, animation and examples.
- *Videos* which provide visual demonstration of the task to be performed as part of transferring practical skills to the students.
- *Circuit Simulation* which provides a comfortable environment for the student to set up the experiment through experiential learning without any fear of damaging the experimental rig and causing injury to himself/herself.

A fault library also will be set up to suggest possible faults with a system by analysing the problem domain and the errors that a student would make. The collected information about expected problems and their solutions, such as how to solve the problem if the motor is unstable or doesn't move, will be used for troubleshooting.

3.5.2 Student Module

There are two major methods suggested in the literature to evolve the student model for the Student Module:

- (a) *Monitoring students' activities:* The students' browsing activities can be monitored by the hypermedia based ITS using passive observations.
- (b) *Assessing students' performance:* Interactive tests or practical measurements can be added to the monitoring tool to provide a better measure of the depth of learning achieved by a student. This approach assumes that access to a page by a student doesn't necessarily mean that he/she has mastered the content of the page.

At this stage of the project, it was decided to implement approach (a) for the student model to speed up the development of a prototype. Further work in the future can extend the method to include assessment as well. Overlay models (as mentioned in 2.4.3) will be used to represent the student's knowledge and learning behaviour. The areas where the student has mastered or has attempted to learn will be recorded. This will be a subset of expert knowledge.

In this laboratory, students work in groups of two. Hence the Student Model for a particular group will be created when the students in the group log into the system for the first time. As the group starts to use CALI, the accessed pages will be recorded in a database. The database will be available to the Laboratory Supervisor. Using the techniques introduced for Web-programming techniques such as Common Gateway Interface, Database and Fill-Out Forms, the monitor can be established with almost no effect on the browsing speed (Rodgers, 1999).

The student's browsing activities contain the exact sequence of the steps taken by each group and provide important information for an intelligent system to guide the students.

3.5.3 Curriculum Module

This curriculum module will include instructional strategies, evaluation methods and presentation control rules as described below:

3.5.3.1 Instructional Strategies

Instructional strategies employed should reflect sound educational and psychological principles satisfying the needs of the students in a particular context. For instance, circuit simulation can be employed to prepare students for setting up a physical circuit when they have developed enough confidence, knowledge and skills. This will prevent possible damage to the equipment or injury to the students due to mistakes made.

According to the requirements of the laboratory, the following instructional strategies will be employed in CALI-II:

- *Tutoring*

The tutor will actively provide domain knowledge in multiple forms such as text, graph, animation and video. For each step of an experiment, the tutor provided sufficient information on “what, when, why and how”, in relationship to any particular topic.

- *Simulation or modelling*

Computer simulations will be provided for critical tasks such as selection of the

equipment or connecting up a particular circuit. The simulator will provide remedial guidance and feedback on the task carried out by the student.

- *Troubleshooting*

The potential problems will be pointed to the students for any particular experiments and possible solutions will be presented through comprehensive text, graphics or animation.

- *Search Engine & Forum*

A search engine will be provided to assist students to search for key words using different logic relations such as “AND” and “OR”. A forum will be used for discussion and posting messages. Any student will be able to post his/her questions to the forum. Instructors or other students will be able to answer the questions. This conversational communication in natural language is convenient for discussion through computer. The problems and their associate solutions posted in a forum will also assist other students to develop a better understanding of the topic.

3.5.3.2 Evaluation Methods

Evaluation is essential for pedagogical (didactic) decision making in order to control content or learning strategy. There are two popular methods of evaluation used in the literature (Warendorf, 1997; Specht, 1998):

- *Traditional evaluation*

The content for traditional evaluation can be easily separated from content for instruction since the former is only for evaluation. The evaluation is performed by comparing the student’s answers against the right answers. Some quizzes in HTML-based forms could be used for a traditional evaluation.

- *Process-based evaluation*

In process-based evaluation, there is no special content specifically designed for evaluation. The evaluation is based on analysing the student's interaction with the system. A full record of the student's activities in the system could provide the basis for a process-based evaluation.

At this stage of the project, a process-based evaluation approach will be employed.

3.5.3.3 Presentation Control Rules

Presentation control organises the sequence of the instructional contents and determines when and where the students' activities should be interrupted to provide feedback, and remedial comments.

If-Then rules can be used for presentation control. "IF" condition corresponds to the student's activity, "THEN" action is the consequent activity taken by the system. For example:

If selection of components is complete and the "Next" button is clicked

Then get URL "xxx.html" for circuit connection

A general form for the above example is:

If the student has finished "Activity 1"

Then start "Activity 2"

Content presentation can be controlled in a predefined mode as the above example or ordered according to different factors as follows:

- *Students' interest*

In order to allow a student to discover proper content according to his/her own interest, a guide or introductory information will be provided for most pages, such as:

If Entering URL “xxx.html”

Then present message “This is a description about xxx. You should master this.

(You haven’t visited it before.)”

- *Students’ activities*

The recorded activities of a student can be used to guide the student to the next appropriate step.

In this work, the second method (Students’ activities) will be used as a reference for guiding the students.

3.5.4 Interface Module

The interface module is a critical aspect of an ITS as it provides the main point of contact between a student and ITS. In this work the Interface Module will be designed to achieve the following objectives:

- **Transparency**

The main objective of the interface design is to make the interface transparent, i.e., to make the domain semantics visible (Burns and Capps, 1988). Hence, the structure of the interface module is similar to that of the expert module with some added facilities such as forum, search engine, note adding and file uploading. Students can view information by navigating through the interface.

- **Consistency**

Consistency ensures that different parts of CALI-II have similar format and behaviour and the package as a whole is homogenous and cohesive. Hence the user will not encounter contradictory concepts to navigate or drive the system when moving from one part to another (Al-Jumeily and Strickland, 1997). Thus

consistent layouts are applied to present information which is of the same type and at the same hierarchical level in Figure 3-5. For example in all the pop-up windows for “What You Should Know”, the positions of the title, instructional contents, control buttons and pop-up menus will remain unchanged. This will assist the user to learn where to look for a particular item or object.

- **Maintaining Students’ Interests**

The Graphical User Interface (GUI) will be designed to provide an attractive and pleasant environment for the user and to maintain his/her attention on the key concepts. The textual representation of information will be improved by adding pictures, animations and video streaming. Video streaming will be also used to demonstrate various practical skills.

- **Ease of Use**

The simplicity of the user interface in terms of its navigation and usage is a critical issue. The interface designed for CALI-II will be simple and natural enough not to introduce additional workload for students to learn it. Students will be able to navigate the page and use its different facilities by relying on their common knowledge and skills of using a Web-browser.

The content of the project is delivered by a standard Web-browser. Hence it should be easily navigable and not require long download time. To reduce download time, authoring tools especially developed for WWW will be used to create small-sized multimedia files. In addition, the CGI language and a high-speed database server will be used as explained later in this chapter.

3.6 Expected Dialogue

CALI-II will be an interactive, multi-functional tutoring system which will respond to various users. Based on the experience gained from CALI-I, the functions expected to be performed by the package and the capabilities offered by a Web-platform, a set of scenarios for the interaction between the package and the users have been specified as part of the design.

It is envisaged that three types of users would interact with CALI-II.

(a) Guest: A Guest is not a serious user of the package but rather a casual visitor.

A Guest could be a future student of the package who is exploring the difficulty level of the course he/she would eventually take. Hence a Guest has a very limited access to the facilities offered by CALI-II. It is unnecessary to record the activities of a Guest or to ask a Guest to provide his/her ID and password.

(b) Student: A Student has enrolled in the subject ELEC344. He/she will carry out all the required experiments. He/she can enrol in the CALI-II system by providing his/her personal identity information and the system will check whether he/she can enrol by comparing his/her login information against the stored subject enrolment information obtained from the Department. The login information is required for recording the browsing activities of the student. A Student can access all the options and student service facilities after a successful login. Two students of the same group can log into the system on the same computer at the same time. From then on, their interaction with the system will be recorded in the database.

(c) Lecturer: The Lecturer is the laboratory supervisor or the course lecturer. He/she can visit all the recorded information such as page access, user information, students' notes and uploaded students' files. These privileges are protected by the Lecturer's ID and password. Hence it is required for the Lecturer to login into the system to use such facilities. The Lecturer's browsing activities are not recorded but his login activities are recorded.

A detailed structure of the system is shown in Figure 3-6 with simulations continued in the Figure 3-7. The first page of the interface is for the user to log in. As mentioned before, a Guest does not need to log in, though he/she can have access to the Laboratory Overview and Report and Assessment details. The same instructions are available for both the Lecturer Access and Student Access but the page access Monitor and Guidance are activated only for the Student Access. The interactions between modules are indicated by the arrowed dotted lines as shown in Figure 3-6. Facilities such as Forum and Search Engine are available for both the Lecturer Access and Student Access, though they have different privileges in managing various facilities. The following privileges are only available for Lecturer Access:

- Deleting messages from the Forum,
- Viewing and changing users' personal information.
- Adding and removing users.
- Viewing and removing groups.
- Viewing page accesses.

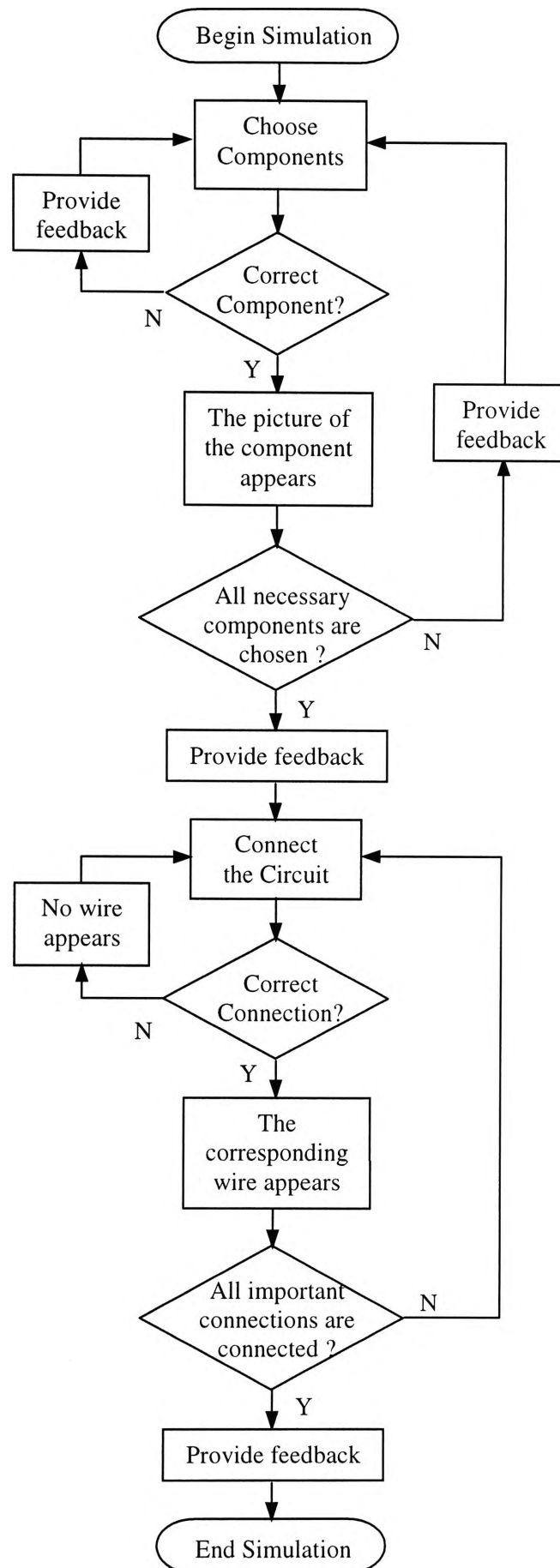


Figure 3-7: Flowchart for Circuit Simulation

On the other hand Students can add, edit or delete their own notes, and upload or delete their files while a Lecturer can only view those notes and files. These functions are disabled in Lecturer Access in order to prevent unexpected changes to students' notes and files. Of course, the Lecturer can edit those notes and files by logging as a student or by accessing the server through FTP or Telnet. Neither Lecturer nor Student can change the records of login and browsing activities through the Interface, but Lecturer or System Administrator can change them by accessing the database in the server.

3.7 Development Platform and Tools

Development of CALI-II with the specifications given in the previous sections will require effective and efficient web-related languages and tools.

After careful study of the specifications defined for the package and design parameters, the following programming tools were selected for the development platform.

3.7.1 Programming Languages

The programming languages primarily employed will include HTML, JavaScript, SQL and PHP3.

PHP3 is a server-side HTML-embedded scripting language which can create CGI style programs to extract form data, to generate dynamic page contents, and to send or receive cookies (PHP: Hypertext Preprocessor, on-line). PHP3 is several times faster than Perl (www.perl.com, on-line) which is another scripting language

commonly used for development of CGI programs. A wide range of databases is supported by PHP3. Writing a database-enabled web page is very simple for PHP3.

In this project, a very fast, multi-threaded, multi-user and robust SQL database server — MySQL will be used. SQL (Structured Query Language) is a standard database language which can easily store, update and access information (MySQL Reference Manual, on-line).

3.7.2 Flash

Taking into account the limited bandwidth of the WWW, Flash (Macromedia Flash, on-line) will be used to develop animations. Flash (originally called FutureSplash) is an animation tool ideal for creating vector-based interactive animations for web pages. At first, a movie is created in the Flash authoring environment. Then its compressed format called Shockwave Flash movies is exported. An HTML document that calls the Shockwave movie is created and both files are uploaded to a web server configured with the proper MIME type that can recognize Flash movies. The movie can be viewed by any web browser equipped with the Shockwave Flash Player. The player is also included in Flash. Flash movies are more versatile than standard GIF pictures while they need less file sizes than GIFs and Director (Macromedia Director, on-line).

In general, Flash has the following advantages:

- **Versatility:** Flash can add interactivity, animations and sound to Web pages. Thus it can create exciting animations, compelling interactivity and wonderful interfaces for Web pages.

- **Simplicity:** In Flash, it is easy to draw graphics, create animations, add interactivity and incorporate sound without doing any scripting. Aftershock utility included with Flash, or an HTML authoring tool such as Dreamweaver can be used to embed the exported movie in an HTML document. Flash can easily complete many functions of dynamic HTML and Applet without programming knowledge of JavaScript and Java.

For Flash 3, JavaScript programs can be used to give more complicated control to the Flash movie but the JavaScript program is outside the Flash movie.

- **Small File Size:** Instead of mapping out images pixel by pixel, Flash creates vector graphics via instructions. This requires less information and thus a smaller file size, which can reduce bandwidth for fast and easy downloading on the Internet.

In addition, scaling a vector-based image will not influence its file size and quality.

- **Browser-independence:** After downloading and installing the Shockwave Flash player (free from the Macromedia web site), the Flash movies look almost the same in different browsers of version 3 or higher. The new versions of browsers bundle the Flash plug-in.
- **Roaming and zooming techniques:** By clicking the right mouse button on a Flash movie and choosing “Zoom In”, the movie can be zoomed in. Without introducing scroll bars, all parts of the zoomed movie can be viewed by dragging it around. Thus very large graphics, especially those with actual size larger than that of the screen, can be displayed in small size and can be viewed with the help of roaming and zooming techniques.

3.7.3 RealSystem

RealSystem and Real Player (Real.com, on-line) will be employed to generate and provide video/audio streaming through Web. In order to avoid long download time, it is necessary to compress the video files or stream them. Streaming allows the audio or video clip to be played as soon as the computer starts to receive or download it. Real System or Real Time Streaming Protocol (RTSP), developed by Real Networks and Netscape Corporation, is a powerful and reliable solution for streaming live audio and video through the Internet or intranets to the web browser with low overhead data delivery. RealPlayer is the most popular player for real-time delivery. It can also play RealAudio and RealVideo files stored locally or temporarily on the computer.

RealAudio or RealVideo can be played from standard Web servers or from RealNetworks RealServer for higher reliability. Its content can be acquired from a variety of sources: pre-recorded or live content from audio and video devices, such as video cameras, microphones, audiotape, videotape and satellite feeds, pre-existing audio and video files, including .wav, .avi, .au and .mov file formats (Powell, on-line). The copyright of the content can be protected as the content provider wishes. Users at different places can view the same video clip at the same time. The clip can be paused, moved forward and back, and started or stopped at any time by a user. The size of the screen can be doubled or even increased to full screen.

3.8 Summary

In this chapter, the basic parameters affecting the design of CALI-II were defined. This included the overall modular structure of the package, the characteristics of the modules and programming platform and tools used in the development of the package.

Chapter 4 : Development of the Interface Module

4.1 Introduction

Development of a CAI system or an ITS requires the undertaking of a number of processes including analysis, design, development, formative evaluation, implementation, summative evaluation, and maintenance. For every hour of computer-assisted instruction, the development time may exceed hundreds of hours. Similar to the design, there are many parameters and issues which should be considered during implementation such as computer platform, programming language and authoring system, achieving an optimised performance, and minimising implementation time and costs. The implemented package should not only be educationally effective but also commercially viable. It should run on affordable computers while providing a high-quality instruction.

In light of the above-mentioned issues, this chapter will primarily report on how the Graphical User Interface (GUI) of CALI-II has been developed. The GUI is the main channel of communication between students and CALI-II. The interface module is an important component in the architecture of such a system to the extent that an ITS can be as intelligent as its interface (Hegarty, 1996).

4.2 Overall Structure and Presenting Methods

Figure 4-1 shows how instructional contents of CALI-II are delivered through the Web-browsers. Different categories of contents are indicated by black text in box while the format in which they are delivered is indicated by grey (or lighter) text outside the box. This structure is the same as that of the expert module.

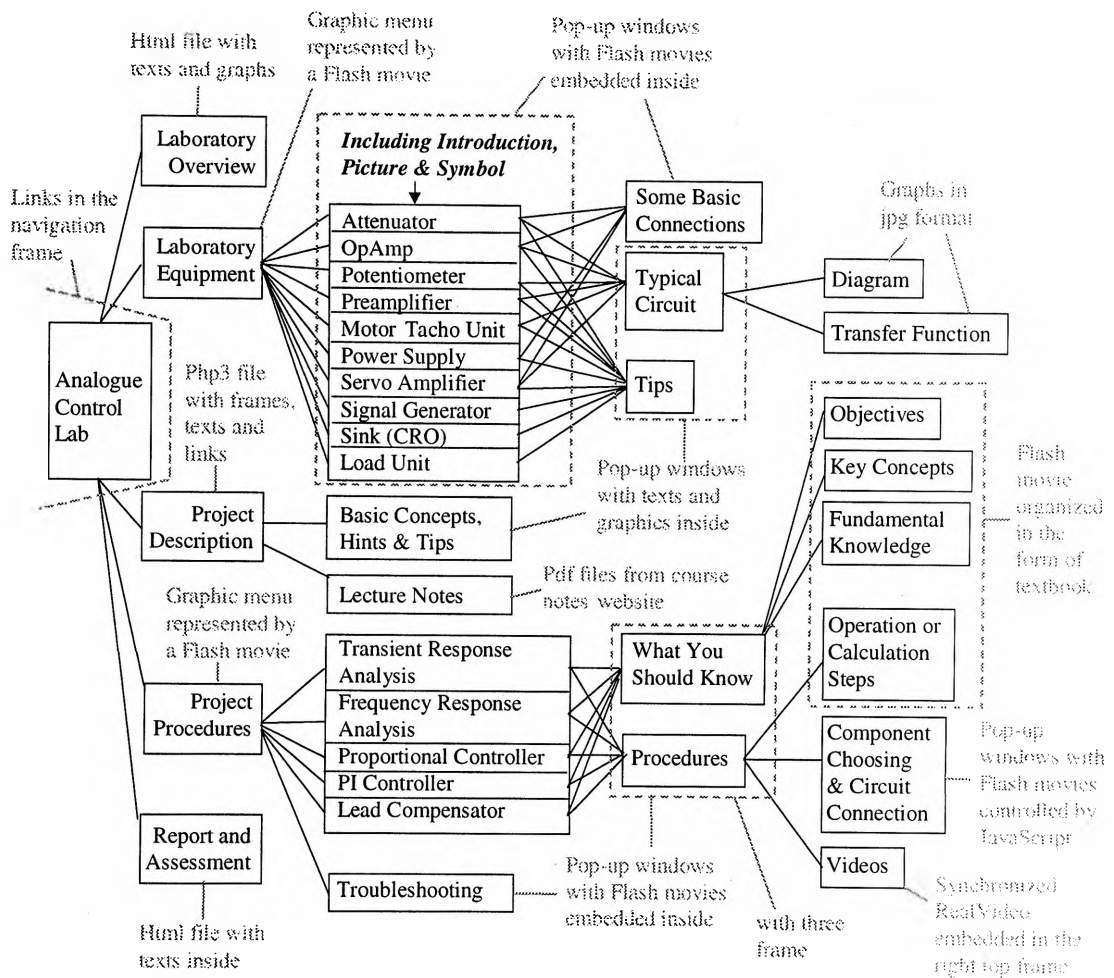


Figure 4-1: Instruction Contents of CALI-II and Delivery Methods

The following hypermedia components are used to make navigation clear and simple to use:

1. Frame and Navigation Frame

Utilising frames in a window, different types of information can be provided in different sections of the same window. For example, in a pop-up window for Procedures, there are three frames displaying a Flash movie, synchronised video and explanatory text supporting the video (Figure 4-2).

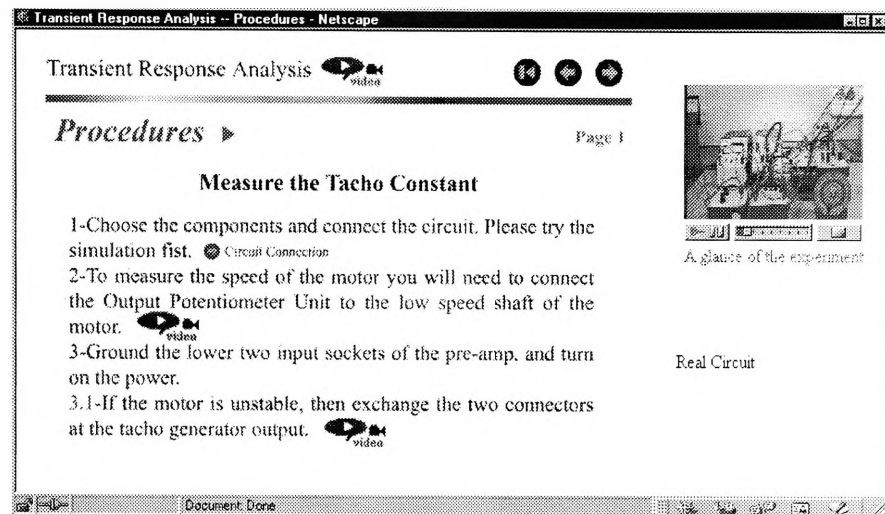


Figure 4-2: A Pop-up Window with Three Frames (the left frame for a Flash movie, the right top one for synchronised video, and the right bottom one for explanatory text supporting the video)

A navigation frame allows the user to easily access the materials linked from the navigation bar. For example, after login, the main page of the CALI-II laboratory appears with a navigation bar frame maintained on the left and a frame on the right for viewing different materials accessed through the links in the navigation frame. A navigation frame is also used in the page for Project Description.

2. Pop-up Window

The links defined in the right frame of the main page of CALI-II leads up to pop-up windows with no tool bar or menu bar, which are obviously different from the main window. If there are many links in the right frame and new information associated with those links is set to be loaded in the same frame, then the user has to go back

and reload the previous content in the right frame again and again. With pop-up windows, the user can view new information without too many disturbances. The size of a pop-up window is smaller than that of the screen. Thus it is almost impossible for a pop-up windows to cover the main window unless they are enlarged. Different contents may use the same pop-up window but with a different title. This will prevent the creation of too many new windows which usually cause confusion to the user.

3. Menu

Menus of different forms are used such as pop-up menus and graphical menus. The pop-up menu related to a button will be hidden if not pointed by the mouse arrow, which makes an effective use of the limited space. The graphical menu will look user-friendly and provide a pleasant interactive environment. A menu can be also designed as the visualized network, with hyperspace nodes and links between them. For example, the menu for Project Procedures can be designed as a network with relationships between the five nodes (corresponding to five experiments respectively) and two nodes in the middle (corresponding to two topics for each experiment).

4. Textbook

Represented in hypermedia form, a textbook will be hierarchically structured into units at different levels such as sections and subsections. There are links inside a textbook, between pages of different textbooks and from the menu to each textbook. For each experiment in Project Procedures, the content of a topic, ie, “What You Should Know” or the operational steps in “Procedures”, is organised as a textbook, divided into pages and marked with page numbers. On each page, there are links to

go to the previous, next and first page, for directly jumping to a subtopic in the same topic or for accessing other information outside this content.

5. *Iconic interaction*

Large graphical displays or icons will be created as buttons or hypernodes to allow students to participate in the domain directly. For example, in the laboratory equipment main page, a graphical menu will be provided with a photo including all the experiment components, in which each component will correspond to an interactive icon or a hyperspace node mapping directly to the detailed information of that component. Students will not have to remember the name of a component to obtain its information. They can immediately access this information by rolling the mouse over the icon and clicking the mouse. This will encourage the students to obtain new information efficiently without getting frustrated and bored.

Multimedia, including pictures, animations and videos, have been used to augment the textual representation of instruction. While multimedia delivers information in a motivational and easy to understand manner, its development is very time consuming and usually troublesome. Additional tools are needed to create and augment multimedia. If the knowledge base is changed, the media may have to be changed while changing or modifying multimedia messages are not as easy as editing a text message.

4.3 Creating and Streaming Videos

In order to demonstrate to the students how the experimental rig can be set up and how it operates, multiple video clips are developed for multiple steps in the procedure and are incorporated in CALI-II.

Before compression and streaming, the videos need to be shot and captured. In this project, a video camera is used to shoot video. Buster video card and its related software — Multimedia Navigator II are used to convert the analogue video into digital format (.avi) (see Figure 4-3). In most cases, a low frame rate of around 10 frames per second is chosen. The quality of each frame is determined by the quality of video capture board. The size of each frame is set to 160×120 pixels, which is a reasonable size for the bandwidth of the Web. Advanced users can customize the video streaming to achieve a specific performance.

Then the (.avi) digital video file is compressed or encoded by RealEncoder (Real.com, on-line) (Figure 4-4). The RealEncoder compresses the audio or video clip by stripping out extraneous layers of information from it. The latest version of RealEncoder can even capture video directly. RealEncoder version 5.0 or higher offers different pre-defined Recording templates, which automatically optimizes RealAudio and RealVideo formats for low-to-high bandwidth options from ordinary modem connections through ISDN, T1 to corporate Intranet LANs. To deliver content over different speed connections from a single web page link, users can choose multiple templates. A “Video High Bit Rate 200k, Voice” was chosen for this project from the pre-defined Recording Templates (Figure 4-3), which almost maximizes the available bandwidth. Thus the quality of the encoded video looks satisfactory. However, the video clip can only be viewed by RealPlayer G2 (Real.com, on-line) or higher and cannot be viewed through a low speed modem since the bit rate is at least 184k even if the video is encoded with mute selected.

RealMedia Tools can be use to cut or augment a video clip (Figure 4-5). The bit rate of a RealVideo clip (.rm) cannot be changed by using the RealMedia Tools. The original .avi

file or videotape will be needed for re-encoding. In the absence of the original video clip, the bit rate of the encoded signal can be reduced by capturing the RealVideo clip as an .avi file, and then using RealEncoder to encode the .avi file at a lower bit rate.

The methods of streaming films with files and synchronising video on the Web can be found in (Blunt, 1998). RealPublisher can also be used to publish RealAudio or RealVideo clip on the Web.

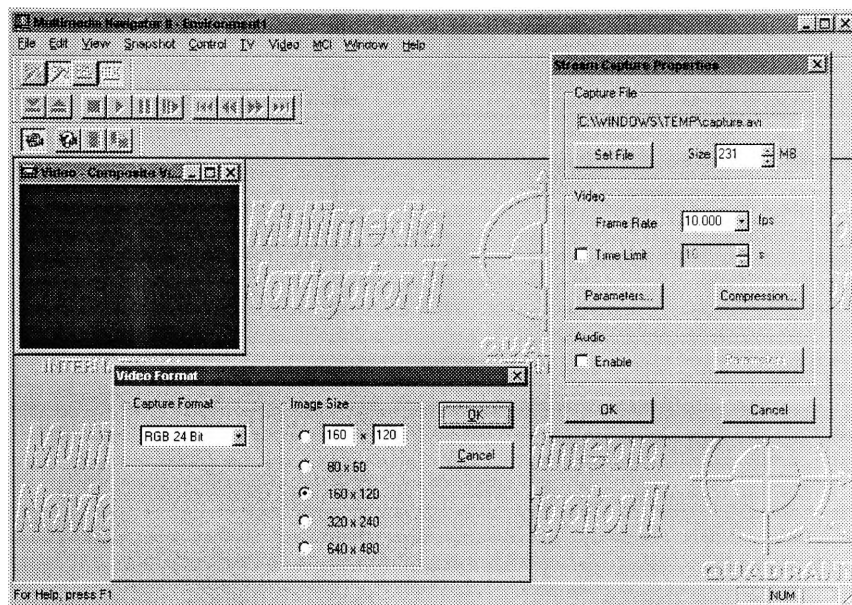


Figure 4-3: Multimedia Navigator II

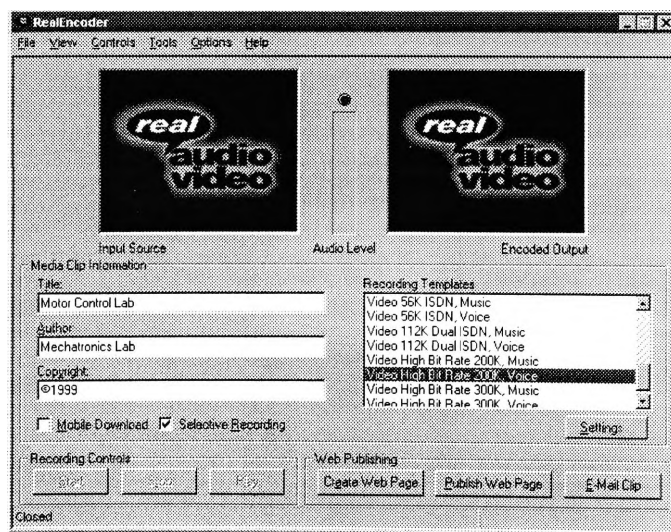


Figure 4-4: RealEncoder (Recording Templates are shown in the right window)

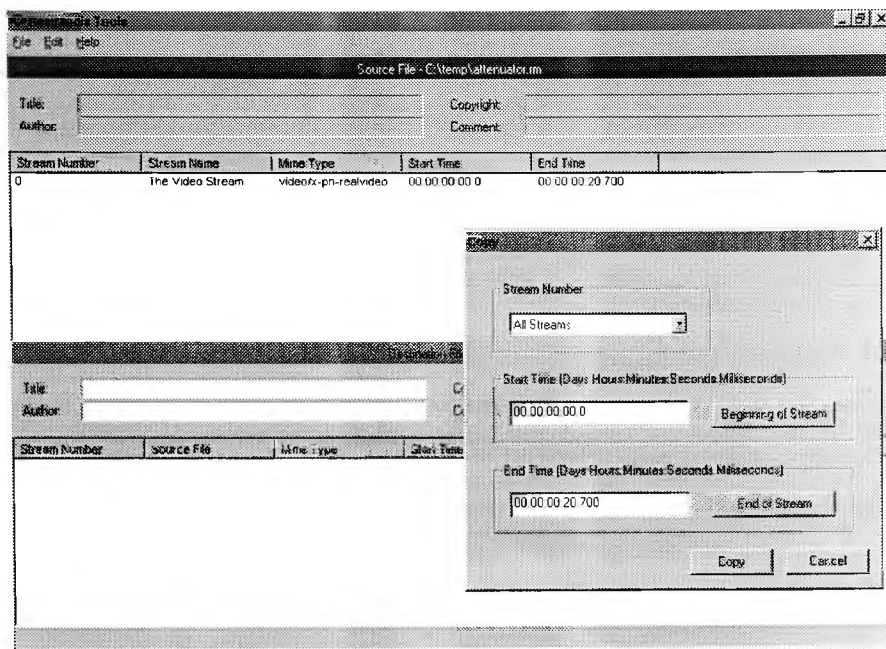


Figure 4-5: RealMedia Tools for Editing .rm Files

4.4 Creating Animation

4.4.1 Flash Movies

There are many sources available on the Internet describing how Flash movies can be created (Michael, on-line; Macromedia Flash — Support Overview, on-line). In this section, the major concepts associated with making a Flash movie, which will be mentioned in this thesis later, are briefly described as follows (Macromedia, Flash Help Pages):

Timeline

The Timeline is used to define the contents of the movie over time. It contains the following components (Figure 4-6):

- *Frames*, which represent a point in time when the movie can change. The number of frames corresponds to the length of time.
- *Keyframes*, are frames with blue dots inside. All changes are made in Keyframes.

- *Layers*, which correspond to the rows in the Timeline. Layers help to organize different parts of a movie and separate elements for animation.
- The Frame and Frame View pop-up menus, which appear when a Frame or the button with a downward pointing triangle is right clicked.
- Onion skin buttons, which control the display and editing of multiple frames on the Stage.
- Timeline header, which indicates the frame numbers, with a pointer pointing to the current frame.
- Timeline status display, which shows from left to right, the number of the current frame, the current frame rate, and the elapsed time to the current frame.

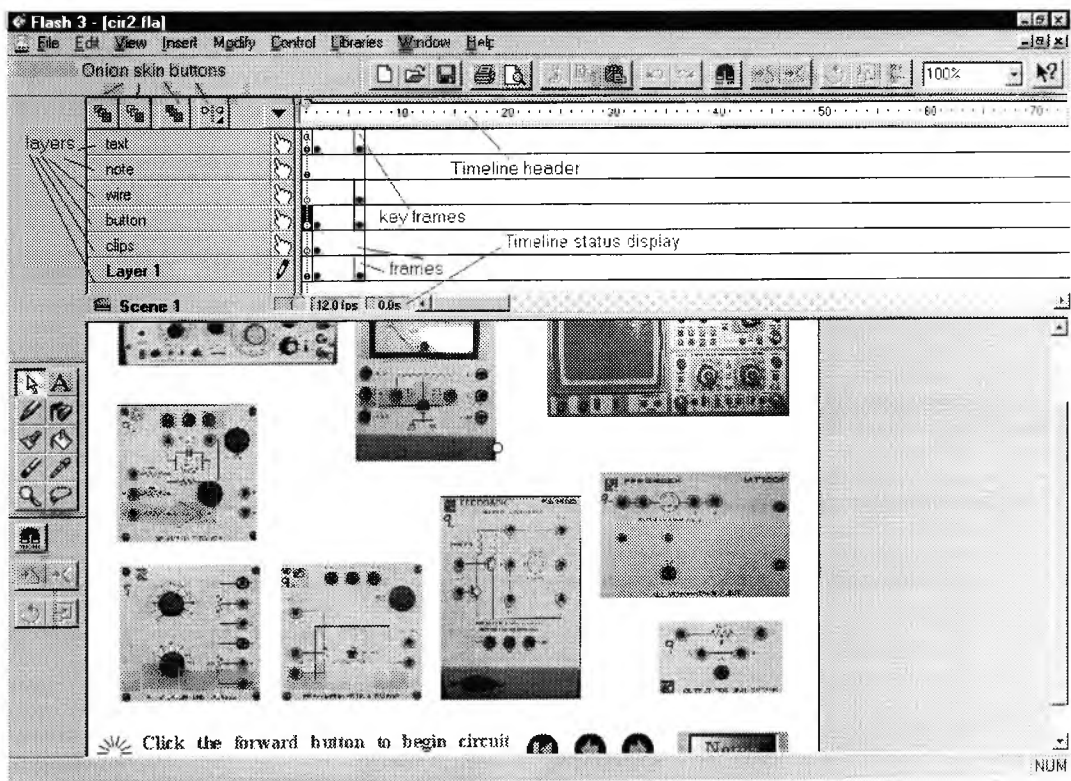


Figure 4-6: Flash Edit Environment with Timeline (Some components of the Timeline are indicated)

Symbol

A reusable object with all the functions of a Flash movie is called a *Symbol*. A symbol may include multiple frames and layers, similar to a scene in a regular movie. In Flash, symbols are stored in a Library (Figure 4-7). Instances of a symbol in time appear as separate elements, but they are linked to a master symbol. Changes made to the master symbol are automatically applied to its instances.

Double clicking the selected instance of a symbol will pop up the Instance Properties dialog box (Figure 4-7), in which instance properties can be changed by assigning actions, or setting special colour effects.

There are three kinds of symbols (Figure 4-7):

- *Graphic symbol*: A graphic symbol has its own Timeline and can contain graphics, even other symbols. It is suitable for the reuse of still images or animations controlled by the main Timeline.
- *Button symbol*: A button symbol has a Timeline with four frames — the mouse-up, mouse-down, mouse-over, and hit states of the button. Actions can be assigned to instances of button symbols.
- *Movie clip*: A movie clip is a completely functional, interactive, and self-contained movie within a movie (Figure 4-8). It plays independent of the main movie Timeline. In a movie clip, all interactive controls work and all sounds play.

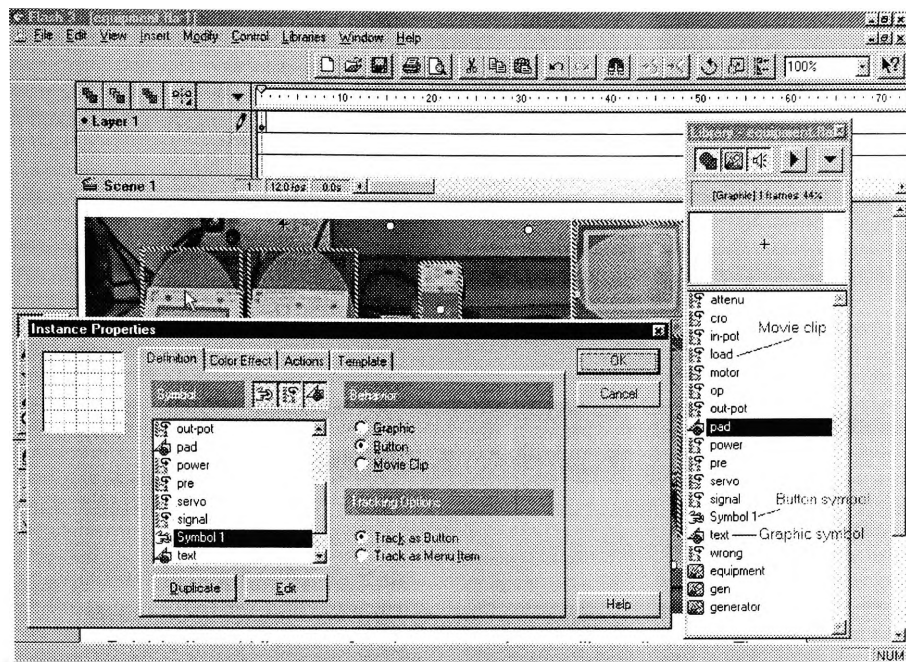


Figure 4-7: Instance Properties Dialog Box (of the Invisible Button on the Power Supply) & Library of Symbols

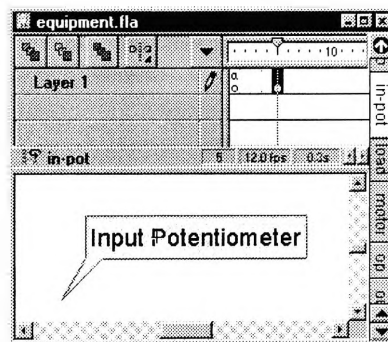


Figure 4-8: Movie Clip in Frame 5 (Note that Frame 1 is empty since the dot in Frame 1 is not filled.)

Actions

There are different types of actions:

- Tell target action (Figure 4-9): This action can be used to control a movie clip. Control actions can be nested within the Begin Tell Target and End Tell Target actions.

- GetURL action (Figure 4-10): This action can load a document from a given URL into the specified window, or call a JavaScript function when scripting methods are used to control the Flash movie.

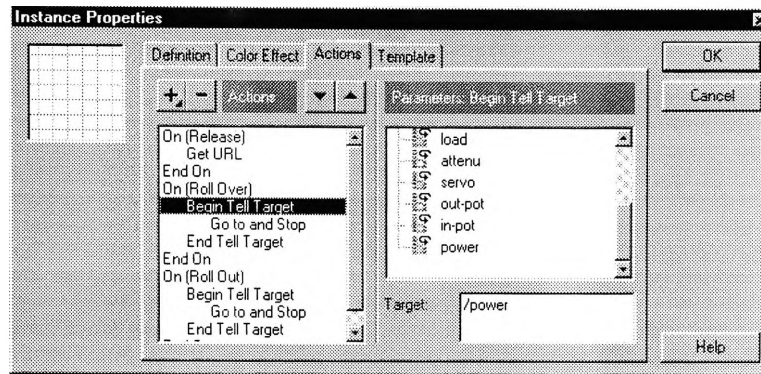


Figure 4-9: Tell Target Action of a Button (Selected target is the movie clip named “power”)

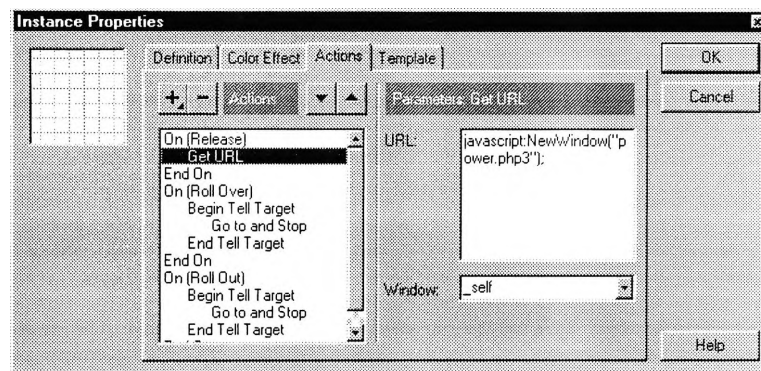


Figure 4-10: GetURL Action of a Button (In the URL area, it’s not an URL but a call of a JavaScript function)

4.4.2 Publishing Flash Movies

The Flash movies developed in this work are embedded into web pages with sizes defined in terms of the percentage of browser window size (Figure 4-11). When a Flash window is enlarged, the movies will still fill the window without losing its quality.

If “list text” is chosen when publishing Flash movies by Aftershock, the texts in Flash can also be embedded in HTML files for search (Figure 4-11).

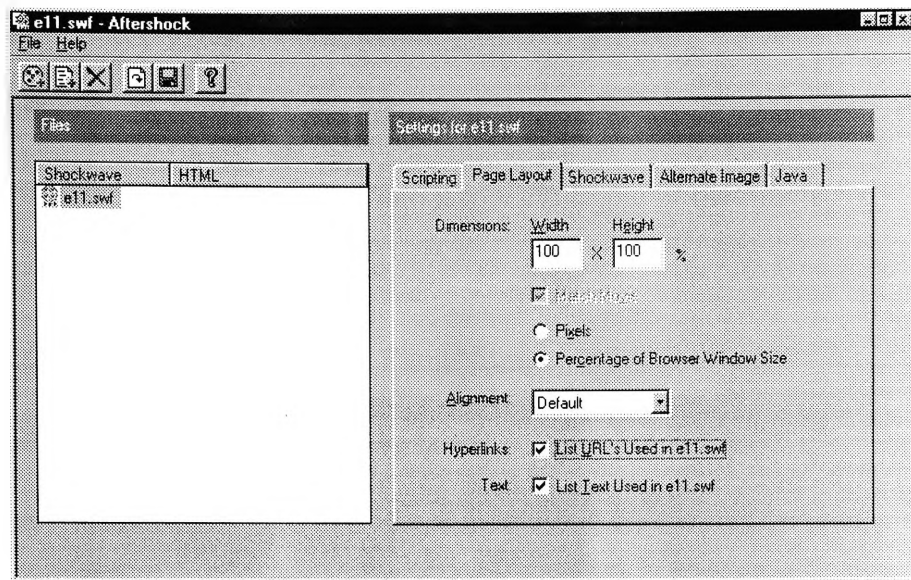


Figure 4-11: Aftershock Publisher (Percentage of Browser Window Size & List Text Used in e11.swf are chosen.)

4.5 Interactive Playground for Circuit Simulation

CALI-II provides an interactive playground for circuit simulation. It provides tools to select the components used in a circuit and connecting the components to form the circuit under the control of the package. In the following sections, creation of this facility will be described.

4.5.1 Control of Flash Movies Using Scripting Methods

Flash 3 supports control of Shockwave Flash movies through JavaScript and the ability to activate JavaScript from .swf (Shockwave) files. This provides new possibilities to produce highly interactive web pages (Macromedia, on-line 1). The Shockwave Flash movies can be activated through many different methods. In this work, the following scripting methods supported by Flash 3 players have been utilised (Macromedia, on-line 1&2):

- `Play()` - Start playing the animation.
- `StopPlay()` - Stop playing the animation.
- `GotoFrame (int frameNum)` - Goto a specific frame of the movie.
- `Rewind()` - Go to the first frame.
- `TGotoFrame (String target, int frameNum)`
- `TGotoLabel (String target, String label)`
- `TPlay (String target)`
- `TStopPlay (String target)`

The “*target*” is the same string used for the Tell Target action.

The above scripting methods have been used in this work to control some interactions and animations in the Flash movies.

4.5.2 Component Selection

The circuit simulation feature of CALI-II has been created by using JavaScript programs and Flash movies. This requires a great deal of effort to design JavaScript programs so that Flash movies could be controlled properly. The JavaScript program for controlling the selection of circuit components in experiment 2 is illustrated in List 4-1.

In this program, JavaScript will not only control a Flash 3 movie to jump to a specific frame of a movie (eg, `map.GotoFrame(6)` in List 4-1), but it can also control a movie clip inside the movie to go to a certain frame (eg, `map.TgotoFrame("_level10/go-on", 2)` in List 4-1).

List 4-1: Comp2.js

```

<!-- Y Chen

//Handle special cases where we need different code
//for Internet Explorer vs. Netscape navigator
var InternetExplorer = navigator.appName.indexOf("Microsoft") != -1;

var num=9;//Set the number of components to be chosen.

//Each component corresponds to an element of MyArray[.]. For
//example, MyArray[1]=1 means the first component has been chosen
//while MyArray[1]=0 means the first component hasn't been chosen
//yet.
var MyArray=new Array(num);

var i=0;
for(i=1;i<=num;i++)MyArray[i]=0; //Set MyArray[.] to be zero.

//Handle the messages sent by the Flash movie
function my_fun(command) {

    //IE and Navigator have slightly different document object models
    //IE treats objects as members of "window" while in Navigator,
    //embedded objects are members of "window.document"
    //For this example "map" is the NAME/ID of the movie we are
    //controlling with the methods available from the plugin or
    //control.
    var map = InternetExplorer ? window.my: window.document.my;

    //If the "back" button is clicked (ie, command="10"), then set
    //MyArray[.] to be zero.
    if(command=="10"){for(i=1;i<=num;i++)MyArray[i]=0;}

    //If the "next" button is clicked (ie, command="0"), the following
    //codes checks whether all the necessary components are chosen,
    //ie, whether no element of MyArray[.] is zero.
    if(command=="0"){

        var c=0;

        //If any element of MyArray[.] is zero, then set the flag c to
        //be 1.
        for(i=1;i<=num;i++){if(MyArray[i]==0)c=1;}

        //If any of the necessary components hasn't be chosen (ie,c=1),
        if(c==1){

            //the movie clip "go-on" (in the first level, ie, level 0),
            //goes to Frame 2
            map.TGotoFrame("_level0/go-on",2);

            //and then the movie clip "go-on" begins to play from Frame 2.
            map.TPlay("_level0/go-on");}

        //If all the necessary components are chosen,
        //the movie goes to Frame 6 which shows that all the //necessary
        components have been successfully chosen.
        else map.GotoFrame(6);

    }
}

```



```
//If the button of a necessary component is chosen, the value of
// "command" will correspond to the order number of the necessary
//component, ie, the value will be neither "0" nor "10".
else{
    i=parseInt(command);

    //The value of the element corresponding to this component is
    //set to be 1.
    MyArray[i]=1;
}
}
//-->
```

If the button of a component not required in a circuit is clicked, a movie clip will indicate that the component is not needed. This interaction is not controlled by the above scripts but by the Flash movie itself.

In the original Flash file, from which the component choosing movie for Experiment 2 is exported, scripts calling the function in the above program are placed in the URL area of the GetURL Action of related buttons. For example, "javascript:my_fun(1):" is placed in that of the "Power Supply" Button (Figure 4-12); "javascript:my_fun(0):" is placed in that of the "Next" Button (Figure 4-13).

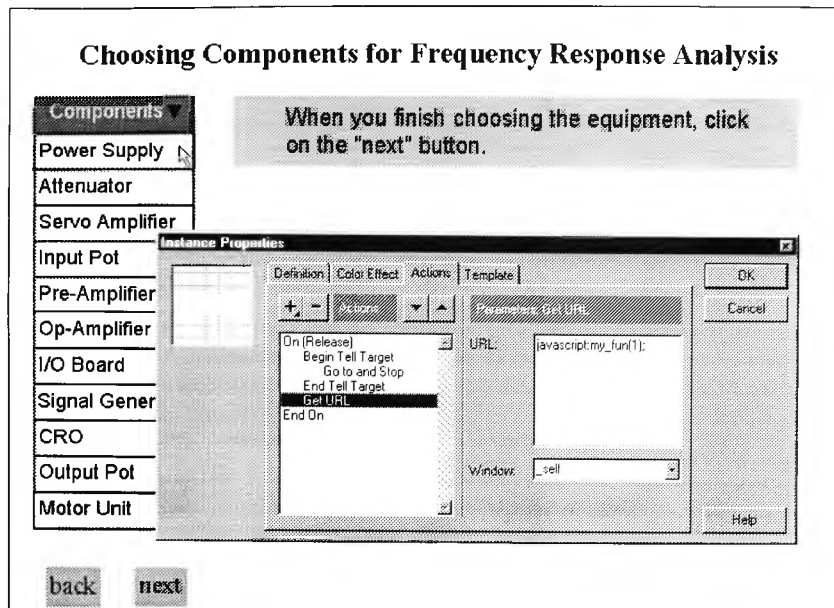


Figure 4-12: Scripts in the URL Area for the Get URL Action of the “Power Supply” Button

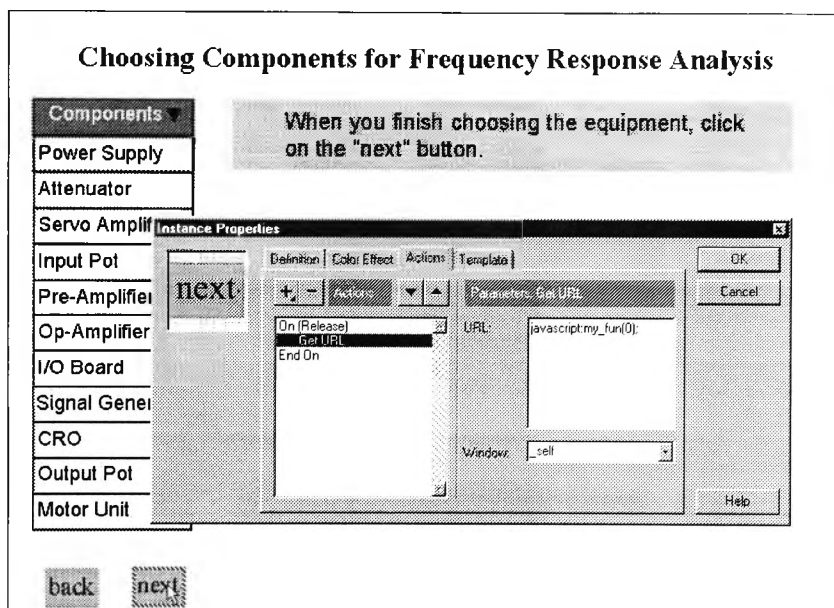


Figure 4-13: Scripts in the URL Area for the Get URL Action of the “Next” Button

The file in List 4-2 is used to display the Component Selection Movie for Experiment 2 on the Web. It allows the communication between JavaScript and the Component Selection Movie. The **SWLIVECONNECT** in EMBED tag must be **true** for this communication to be available in the Netscape browser.

List 4-2: /exp2/ex3.php3

```

<html>
<HEAD>
<TITLE>Transient Response Analysis -- Component Choosing</TITLE>
<SCRIPT LANGUAGE="JavaScript" SRC="comp2.js">
</SCRIPT>
</HEAD>
<!--Y Chen-->
<BODY bgcolor="#FFFFFF">
<?      include('../..//monitor/include/recorder1.inc');?>
<OBJECT CLASSID="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
CODEBASE="http://active.macromedia.com/flash2/cabs/swflash.cab#vers
ion=3,0,0,0"
        ID=my WIDTH=100% HEIGHT=100%>
        <PARAM NAME="movie" VALUE="comp2.swf">
        <PARAM NAME="play" VALUE="false">
        <PARAM NAME="quality" VALUE="high">
<EMBED NAME=my SRC=comp2.swf WIDTH=100% HEIGHT=100% PLAY=false
        SWLIVECONNECT=true QUALITY=high>
        </OBJECT>
</body>
</html>

```

4.5.3 The Simulation for Circuit Connection

Using communication between JavaScript programs and Flash movies, circuit connection simulation has been realized. If and only if two terminals of a potential link between two components are successively clicked by the user, a wire connects the two terminals using a Flash movie. One terminal of a connection may correspond to more than one socket on a component since different sockets may be connected by a shortcut inside the component. Clicking any socket belonging to the terminal means clicking this terminal. If between the above two successive clicks, there is any other click at other locations, the wire will not appear. This makes it hard for a student, who may know nothing about the circuit, to get a right connection by clicking randomly.

The JavaScript program given in List 4-3 controls the movie connecting the circuit in Experiment 2. The parameter “num” contains the number of wires which should be connected by the student.

List 4-3: Cir2.js

```

<!-- Y Chen

//Handle special cases where we need different code
//for Internet Explorer vs. Netscape navigator
var InternetExplorer = navigator.appName.indexOf("Microsoft") != -
1;

var num=10; //Set the number of important wires to be connected.

//Each important wire corresponds to an element of the array c[].
//For example, c[1]=1 means the first important wire has been
//connected. while c[1]=0 means the first important wire hasn't
//been connected yet.
var c=new Array(num);

var OK=1; //Set the flag OK to be 1.

var i=0;
for(i=1;i<=num;i++){c[i]=0;} //Set c[.] to be zero.

//Handle the messages sent by the Flash movie
function my_fun(command,arg) {

//IE and Navigator have slightly different document object models
//IE treats objects as members of "window" while in Navigator,
//embedded objects are members of "window.document"
//For this example "map" is the NAME/ID of the movie we are
//controlling with the methods available from the plugin or
//control.
var map = InternetExplorer ? window.my: window.document.my;

//If the "back" button is clicked (ie, command="50"), then set
//c[.] to be zero and return from this function.
if(command=="50"){for(i=1;i<=num;i++)c[i]=0;return;}

//If the two terminals of an important connection are clicked
//successively, the value of "command" will correspond to the
//order number of this important connection, ie, the value will
//be neither "50" nor "100", the corresponding element of c[.]
//will be set 1, which means this connection is successfully
//connected.
if(command!="50"&&command!="100"){i=parseInt(command);c[i]=1;}

OK=1;

//If any of the important connections hasn't be connected, the
//flag OK is set to be 0.
for(i=1;i<=num;i++){if(c[i]==0)OK=0;}

//If all the important connections are connected, the movie goes
//to Frame 6 which shows all the connections and the message of
//"Well done!".
if(OK==1){map.GotoFrame(6);}

//If a valid socket button or invalid places is clicked (ie,
//command="100"),
if(command=="100"){

//check all important connections.

```

```

for(i=1;i<=num;i++){

    //If an important connection hasn't been connected (ie,
    //c[i]=0) and the current click is not a valid click (ie,
    //parseInt(arg)!=i)
    if(c[i]==0&&parseInt(arg)!=i){

        //The movie clip "wirei", which corresponds to the ith
        //important connection in level 0, goes to Frame 1. This
        //insures that a wire will not appear if two terminals of
        //this important connection are not successively clicked.
        string="_level0/wire"+i;
        map.TGotoFrame(string,1);
    }
}
}
}
//-->

```

Figure 4-14 shows the Frame 2 in Scene 1 of the Flash file from which the circuit connection movie for Experiment 2 is exported. The small round buttons on the sockets, which make the sockets a little unclear in the Figure 4-14, are invisible in the exported movies, i.e, they will not affect the appearances of the sockets. Each connection or wire together with its terminal buttons is included in a movie clip. Figure 4-15 shows the first frame of the movie clip “wire9” in this Flash file. For the Get URL Action of the invisible button pointed to by the mouse arrow, the scripts in the URL area are “javascript:my_fun(100,9);”. The figure “100” indicates that the button is a connection terminal. The figure “9” indicates that the connection is numbered “9” and is in “wire9” movie clip. Actions of buttons “Terminal 1” (the upper button) and “Terminal 2” (the lower button) for all keyframes of Movie Clip “wire9” are shown in Table 4-1 (N/A means this action is not used). Figure 4-16 shows how Movie Clip “wire9” is controlled. The rhombus is used in Figure 4-16 to mean that a click is met and the condition is checked. In Frame 10, there is no button except the wire connecting the two terminals. i.e., the 9th wire is successfully connected.

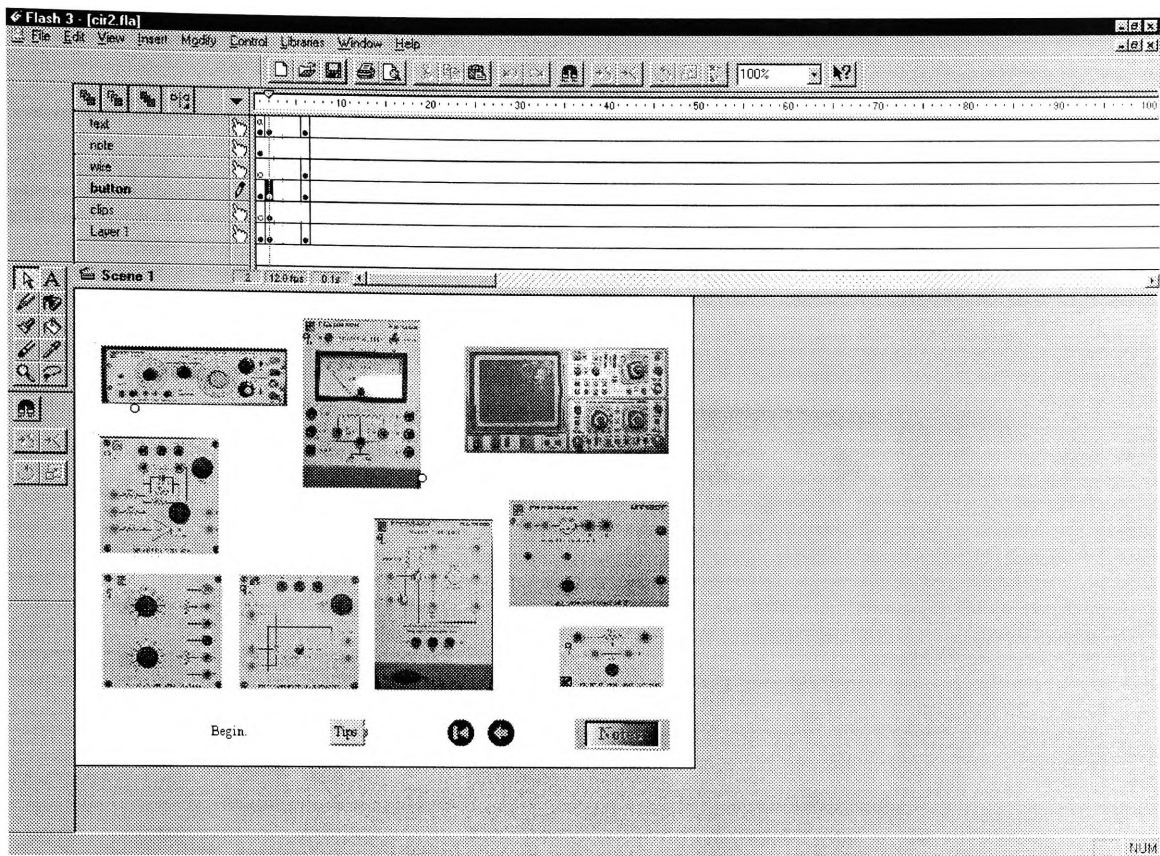


Figure 4-14: Frame 2 in the Scene

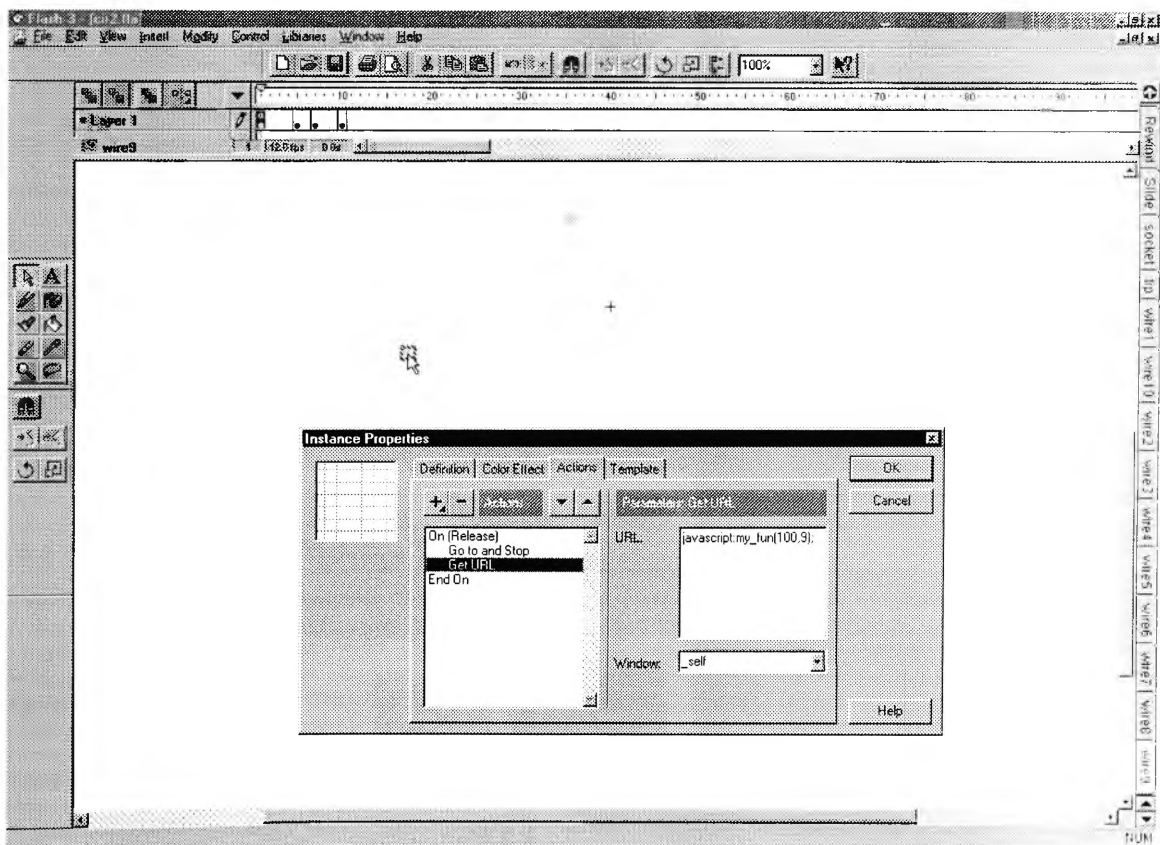


Figure 4-15: Movie Clip "wire9"

Table 4-1: Actions of Buttons “Terminal 1” and “Terminal 2” in Keyframes of Movie Clip “wire9”

Frame	Terminal 1		Terminal 2	
	Go to and Stop (Frame)	Get URL	Go to and Stop (Frame)	Get URL
1	5	javascript:my_fun(100,9);	7	javascript:my_fun(100,9);
5	5	N/A	10	javascript:my_fun(9);
7	10	javascript:my_fun(9);	7	N/A

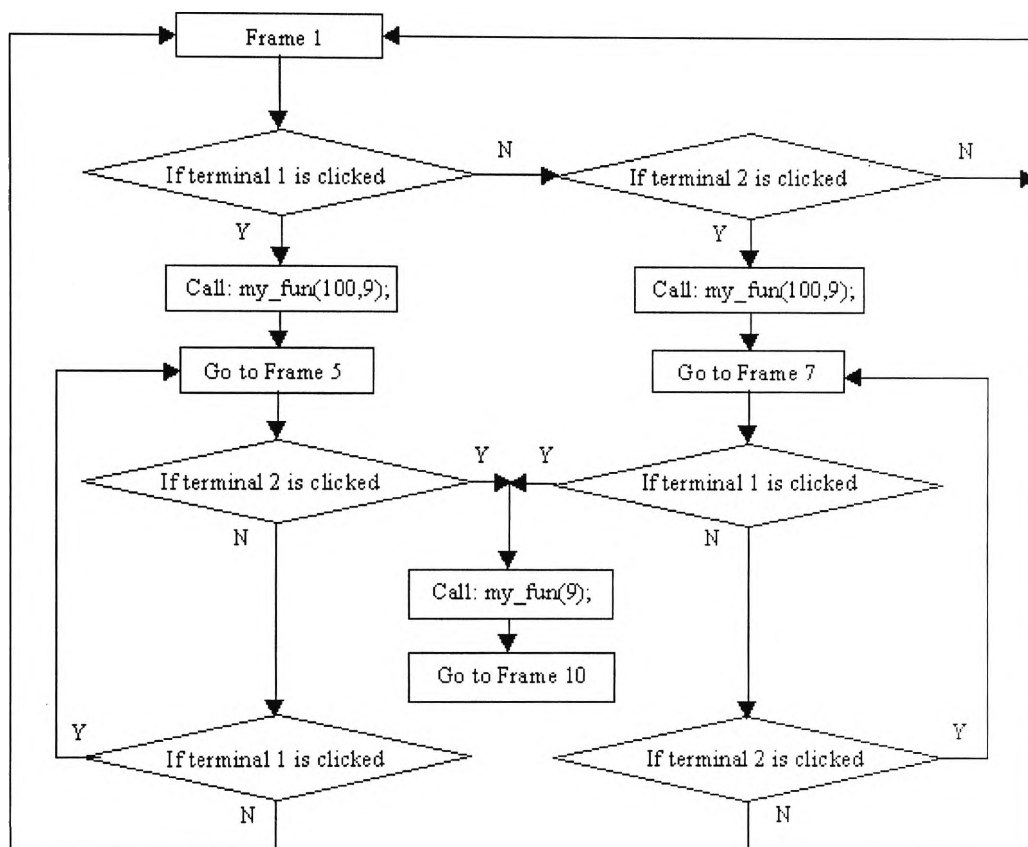


Figure 4-16: Diagram Explaining How Movie Clip “wire9” Is Controlled

4.6 Creating the Pop-up Windows

JavaScript is used to make a button (in a Flash 3 movie) pop up a window without the tool bar and the menu bar. For example, for the invisible oval button on the Power Supply, “javascript:NewWindow(“power.php3”);” is put into the URL area of the

GetURL Action of the invisible Button named “Power Supply” (Figure 4-17). The methodology used is shown in JavaScript program “Equ.js” that defines the properties of the pop-up window (List 4-4). This is included in the web page for “Laboratory Equipment” (List 4-5).

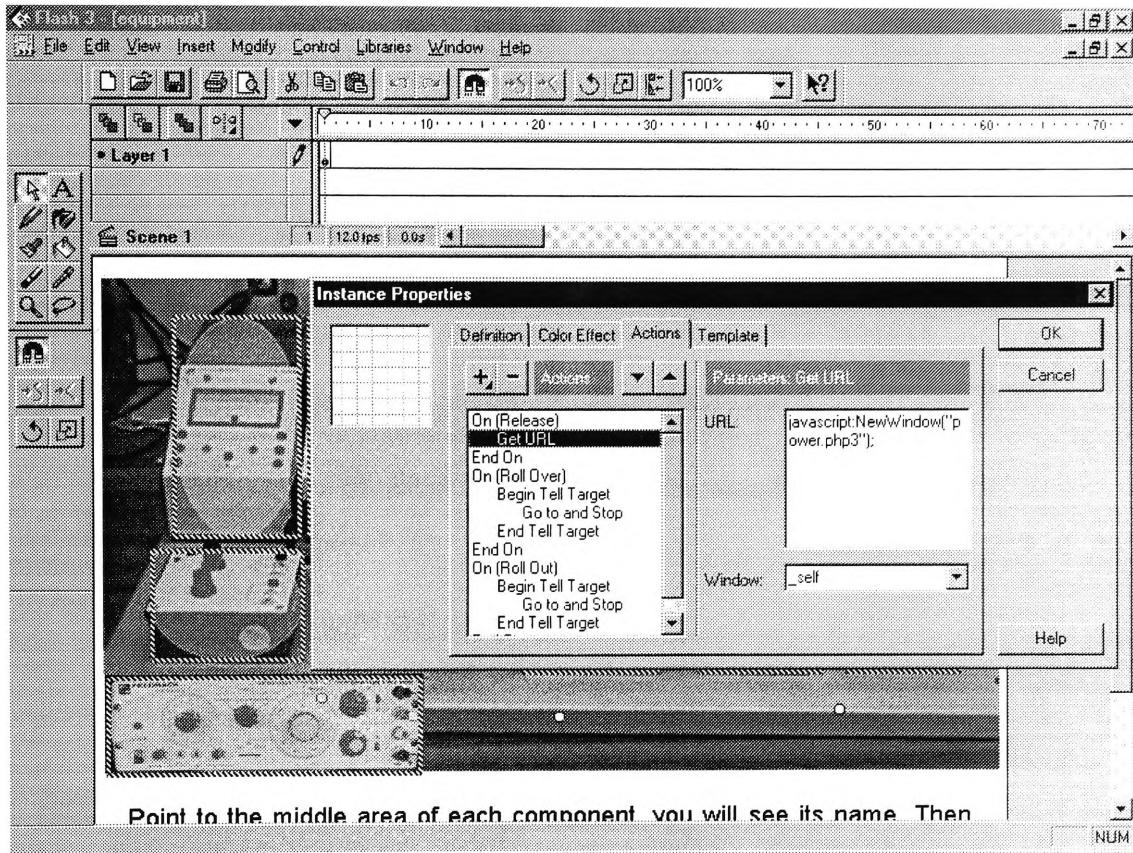


Figure 4-17: Scripts in the URL Area for the Get URL Action of the Invisible Button on the Power Supply

List 4-4: Equ.js

```
<!--Y Chen
function NewWindow(content) {
    TipsWin= open(content, "NewWindow",
        "width=640,height=450,status=yes,toolbar=no,resizable=yes
,menubar=no");
}
// -->
```


List 4-5: Equipment.php3

```

<HTML>
<HEAD>
<TITLE>Equipment</TITLE>
<SCRIPT LANGUAGE="JavaScript" SRC="equ.js">
</SCRIPT>
</HEAD>
<BODY bgcolor="#FFFFFF">

<!-- Y Chen -->
<?      include('../monitor/include/recorder.inc');?>
<OBJECT classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"

codebase="http://active.macromedia.com/flash2/cabs/swflash.cab#vers
ion=3,0,0,0"
  ID=equipment WIDTH=100% HEIGHT=100%>
  <PARAM NAME=movie VALUE="equipment.swf">
  <PARAM NAME=quality VALUE=autohigh>
  <PARAM NAME=bgcolor VALUE=#FFFFFF>
<EMBED SRC="equipment.swf" swLiveConnect=FALSE WIDTH=100%
HEIGHT=100%
  QUALITY=autohigh BGCOLOR=#FFFFFF TYPE="application/x-shockwave-
flash"

PLUGINSOURCE="http://www.macromedia.com/shockwave/download/index.cgi
?P1_Prod_Version=ShockwaveFlash">
</EMBED></NOEMBED></OBJECT>
</BODY>
</HTML>

```

4.7 Summary

This chapter described the implementation of the Interface Module with emphasis on delivering instructions, creating videos, animations, simulations and pop-up windows. The building of intelligence into the system, which especially lies in the Student Monitor and Guidance, will be explained in next chapter.

Chapter 5 : Student Monitor & Guide

5.1 Introduction

The creation of a Student Monitor and Guide in CALI-II makes the package more intelligent. Student Monitor, as an important part of the Student Module, constantly monitors the students' activities and responses. It sets up and keeps updating the dynamic Student Model, according to which the information content and feedbacks are provided or modified in the best possible way to maximise students' learning capabilities (Rickel 1989). Guidance of students' activities is provided by the Tutor Module and can be changed if different rules are applied or added.

This chapter provides an overview of these two facilities while highlighting their important features.

5.2 Student Monitor

Student Monitor adds a user to the "user" table in the database if a student successfully enrolls in the laboratory. For each user, the "user" table records the user's identification code (or the student's student no.), first name, surname, password, group number, user type (ie, a lecturer or a student) and email address. The navigation activities are recorded if the user of the system is a student or the users are in the same group. When a student (or two students in the same group) enters the identification code and password, the system will search the "user" table and offer the students the ability to login. The system will not allow the student to proceed if the identification code and password have not been entered correctly. The student

number is used as the identification code. Although a student may modify his/her password and notes, his/her activities and attendance may not be changed or deleted by the student. The functional procedure of the Student Monitor at the login stage is illustrated in Figure 5-1.

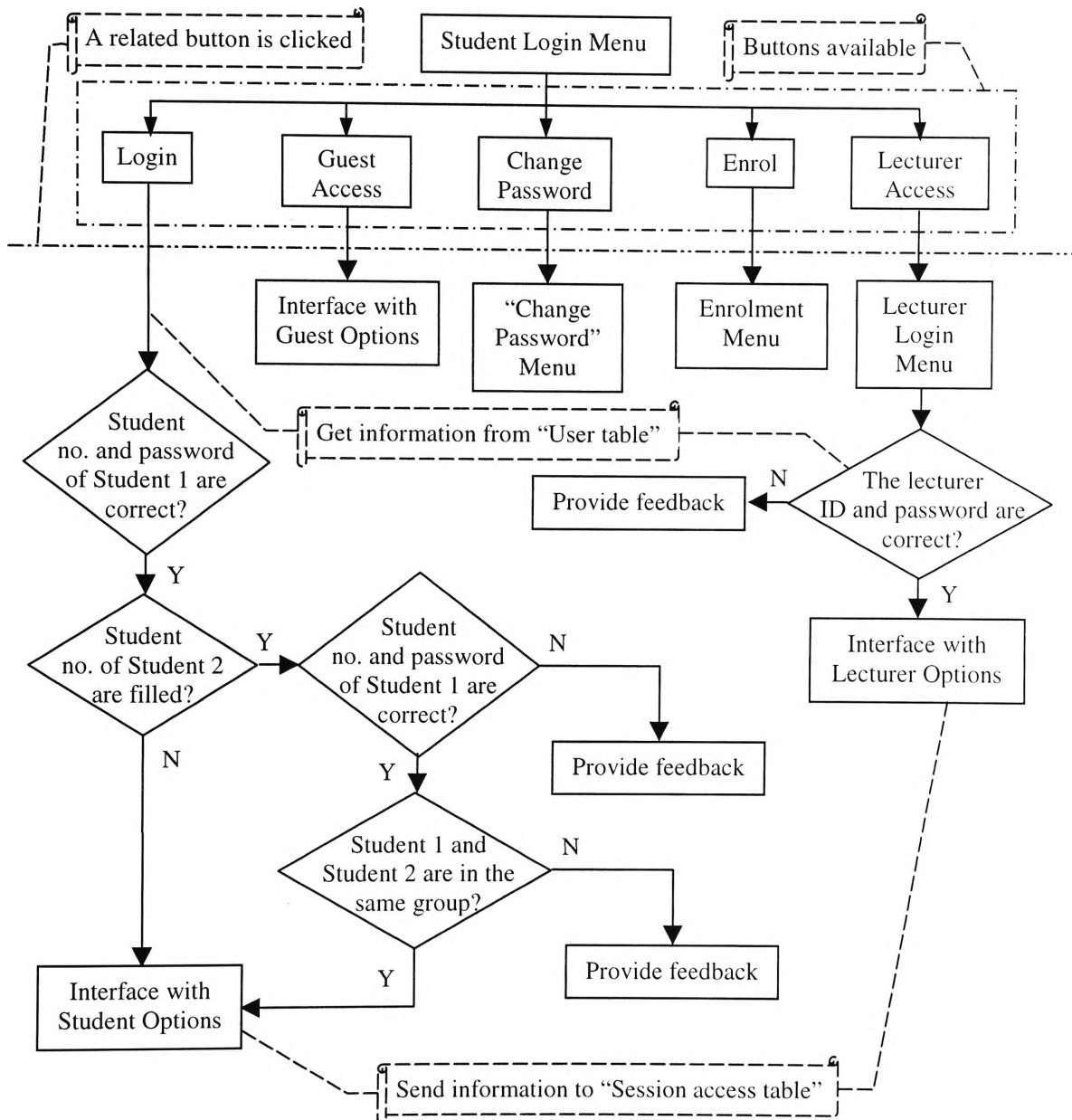


Figure 5-1: User Authentication

The enrolment form or menu (shown in Figure 5-2) is for students to enrol by themselves. After students fill the form and click the "Enrol" button, the PHP3 file

checking the enrolment will run. Figure 5-3 indicates how the enrolment is checked and processed. "Student.txt" is obtained from the department database and stored in the system, in which each line of the main text body lists the course name, the student number of a student, the student's name and other information. After the system determines that the student or two students can be enrolled, the subfunction for adding a user (whose diagram is shown in Figure 5-4) will be called (Rodgers, 1999). If one student successfully enrolls himself or two students successfully enrol at the same time, the system will assign a group number automatically. The first enrolled group will be assigned as Group no. 1.

Enrolment Menu

	Student 1	Student 2
Student No.:	<input type="text"/>	<input type="text"/>
Surname	<input type="text"/>	<input type="text"/>
Given Names	<input type="text"/>	<input type="text"/>
Email Address	<input type="text"/>	<input type="text"/>
Password	<input type="text"/>	<input type="text"/>
Confirm Password	<input type="text"/>	<input type="text"/>

Figure 5-2: Enrolment Form

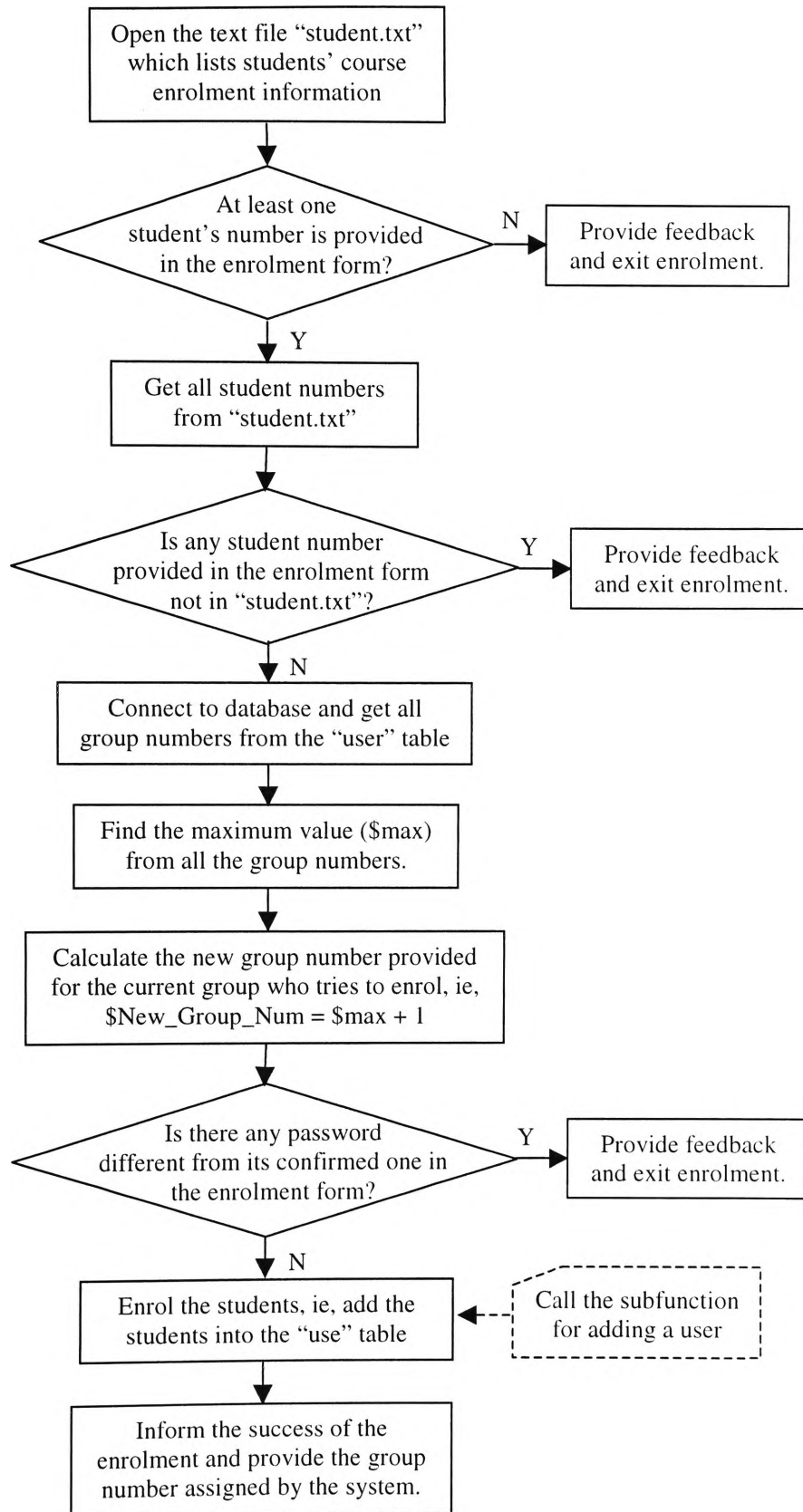


Figure 5-3: Procedure in Checking the Enrolment

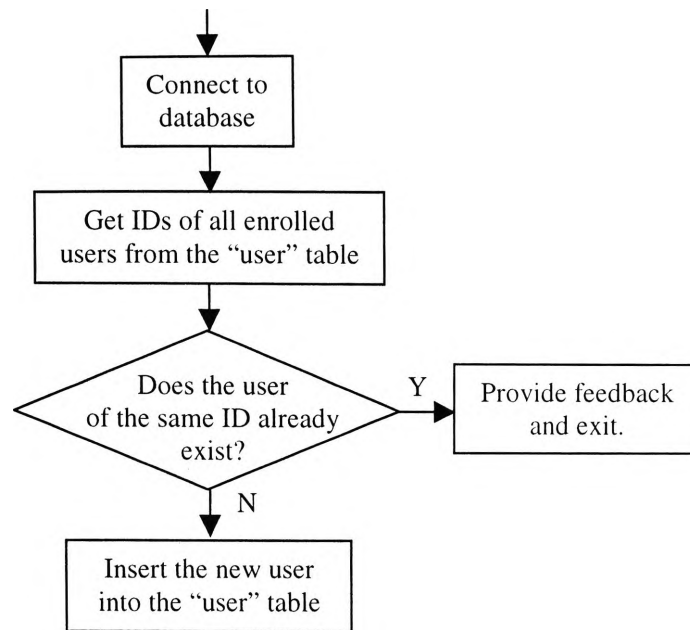


Figure 5-4: Procedure for Adding a User

The supervisor can add a student to the database in case the student has problems in enrolling himself/herself. By logging into Lecturer Access, the supervisor can add any user as a Lecturer or as a Student to any group (Figure 5-5), change their details (Figure 5-6) or remove them from the database (Figure 5-7) (Rodgers, 1999).

Add User

Student Number/ID <input type="text"/>	User Type <input type="text" value="student"/>
Surname <input type="text"/>	Password <input type="text"/> Group Number <input type="text"/>
Given Names <input type="text"/>	Email <input type="text"/>
<input type="button" value="Add User"/>	

Figure 5-5: Menu for Adding a User

Change User's Details

Student Number/ID to change <input type="text"/>	User Type <input type="text" value="student"/>
Surname <input type="text"/>	Password <input type="text"/> Group Number <input type="text"/>
Given Names <input type="text"/>	Email <input type="text"/>

Figure 5-6: Menu for Changing a User's Details

Remove User

Student Number / ID to remove

Figure 5-7: Menu for Removing a User

The Student Model developed in this work records various aspects of students' or groups' activity. Students' records are stored in separate tables in the database. They include tables for students' personal information (Table "user"), students' notes (Table "notebook"), students' uploaded files (Table "reports"), students' login information (Table "session_details"), and page access information (Table "page_accesses"). For each entry, the login information indicates the group number, the IDs of the students in the group, the date, the log on time, the log off time and the computer IP address. For each access of a file, the page access information indicates the group number, the file name, the access date and time. The student monitor constantly updates the student's login information and page access tables (Rodgers, 1999). The records of these two tables will always exist on the system unless deleted by the administrator.

At the end of each session, the instructor can check the student records to check the attendance of a student or the hit frequency of a page. If the hit frequency of a device is very high for all the students (or a particular group), it may indicate that all the students or a particular group had problems using that device. The records are also used by the system to provide dynamic feedback to the students.

The recorded page access of a group can be checked by the tutors at any time. Figure 5-8 shows the login records viewable from the interface for the Lecturer Access. If only one student is logged in, the column of another student would be empty. After a group number is clicked (Figure 5-9), the right frame shows all the usage logs of the group (Figure 5-10). If the link of a date is clicked (Figure 5-11), the group's page access logs on that date will appear (Figure 5-12).

Class Access Statistics						
Group	Student1	Student2	Date	Logon Time	Log Off Time	Remote Address
	c		15/10/1999	19:22:24		130.130.68.5
	c		15/10/1999	19:22:46		130.130.68.5
	c		15/10/1999	19:26:00		130.130.68.5
	c		15/10/1999	19:34:07		130.130.68.5
	c		15/10/1999	19:50:20		130.130.68.5
	c		15/10/1999	20:22:23	20:39:25	130.130.68.5
	c		16/10/1999	15:04:37		130.130.68.5
10	c		13/10/1999	19:21:12		130.130.68.5
10	c		13/10/1999	19:25:58		130.130.68.5
10	c		13/10/1999	19:38:23		130.130.68.5
10	c		13/10/1999	19:41:42		130.130.68.5
10	c		13/10/1999	19:45:11		130.130.68.5
10	c		13/10/1999	19:51:57		130.130.68.5
10	c		13/10/1999	19:52:34		130.130.68.5
10	c		13/10/1999	19:54:11		130.130.68.5
10	c		13/10/1999	20:01:35		130.130.68.5
10	c		14/10/1999	13:52:54		130.130.68.5
10	c		14/10/1999	13:52:59		130.130.68.5
10	c		14/10/1999	13:56:10		130.130.68.5
10	c		14/10/1999	13:57:59	14:34:22	130.130.68.5

Figure 5-8: Class Accesses or Login Records

34	9829855	9830116	13/03/2000	09:19:13		130.130.88.100
34	9829855	9830116	13/03/2000	09:23:18		130.130.88.100
35	9835991	9461337	14/03/2000	10:02:06	12:04:44	130.130.88.100
35	9835991	9461337	14/03/2000	10:57:08		130.130.88.231
36	9822668		14/03/2000	10:07:58		130.130.88.236
36	9822668		14/03/2000	10:15:32		130.130.88.236
36	9730031		14/03/2000	10:14:30		130.130.88.236
37	9730031		14/03/2000	10:15:38		130.130.68.19
5	9577386	9810575	14/03/2000	10:01:45		130.130.88.100
6	9831934	9831941	13/03/2000	10:35:38	10:37:06	130.130.159.123
6	9831934	9831941	13/03/2000	14:24:08	15:42:20	130.130.158.63
6	9831934	9831941	14/03/2000	17:14:06		130.130.158.107
7	9820254		14/03/2000	13:01:19	13:11:59	130.130.157.59

Figure 5-9: Clicking a Group Number

Usage Log of Group 36

Student1	Student2	Date	Logon Time	Logoff Time	Remote Address
9822668		14/03/2000	10:07:58		130.130.88.236
9822668		14/03/2000	10:15:32		130.130.88.236

Figure 5-10: Usage Log (clicking the link of a date)

Page Access Log for Group 36 on 14/03/2000

Filename	Open Time
menu.php3	10:08:18
project-description.php3	10:08:34
menu.php3	10:08:46
/exp1/ex2.php3	10:08:51
/exp1/ex3.php3	10:15:58
/exp1/ex3.php3	10:16:13
menu.php3	10:18:18
/exp1/ex2.php3	10:18:34
/exp1/ex3.php3	10:21:47
cir1.php3	10:22:09
trouble.php3	10:41:04

Figure 5-11: Page Access Log on a Date

5.3 Feedback Provided by the Tutor Module

A typical feedback provided by the Tutor Module using predefined rules is illustrated in this section. The codes for guidance can be inserted into the codes for page access recordings. For example, it is expected that all the students should study all the devices before connecting a circuit. The groups are also expected to go through different milestones in the order defined by the instructor. Otherwise students will have difficulty in either setting up the experiments or achieving the objectives set for them.

In order to maintain a list of such preferences, a table named “files” is created in the database (Figure 5-12) keeping the names and order of the files to be browsed. Each file in the table is assigned a file number indicated by the number in Column “nf”. Another table called “visiting” is created to store the hits of these files for each group with initial values set to be 0 (Table 5-1). The term “f1” refers to the first file in “filename” column of Table “files” (ie, attenu.php3); “f2” means the second file (cro.php3) and so on. The parameter “ng” refers to the group number (Assume that there are only six groups).

```
mysql> select * from files;
```

nf	filename	forwhat
1	attenu.php3	Attenuator
2	cro.php3	Osilloscope or Sink or CRO
3	load.php3	Braker or Load Unit
4	motor.php3	Motor Tacho
5	op.php3	Operational Amplifier
6	out-pot.php3	Out Potentiometer
7	power.php3	Power Supply
8	pre.php3	Pre-Amplifier
9	servo.php3	Servo-Amplifier
10	signal.php3	Signal Generator or Source
11	cir1.php3	circuit connection for Experiment 1
12	cir2.php3	circuit connection for Experiment 2
13	cir3.php3	circuit connection for Experiment 3
14	cir4.php3	circuit connection for Experiment 4
15	cir5.php3	circuit connection for Experiment 5

```
15 rows in set (0.00 sec)
```

Figure 5-12: Contents in the “files” Table

Table 5-1: Contents in the Initial “visiting” Table

ng	f1	f2	f3	f4	f5	f6	f7	f8	f9	f10	f11	f12	f13	f14	f15
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The codes indicating the preferred navigation route are then added to the file “recorder.inc” and “recorder1.inc”. List 5-1 shows that in “recorder1.inc” the added codes are between star lines with the title “Guidance” as the following:

```
//Guidance*****
```

.....

```
//*****
```

The file “recorder.inc” or “recorder1.inc” is included at the beginning of every page whose access is required to be recorded (PHPBuilder.com, on-line; Rodgers, 1999). It will check the access permission and will record students’ page access to control the navigation order. “Recorder.inc” is similar to “recorder1.inc” except that the codes between the lines:

```
//=====
```

are changed to:

```
$filename = basename(getenv("REQUEST_URI"));
```

and the line:

```
include('../../monitor/include/connect.inc');
```

is change to

```
include('../monitor/include/connect.inc');
```

which means that “recorder.inc” is included in files in directories at the same level as the directory \monitor while “recorder1.inc” is included in files in directories at the

lower level. The diagram of the “recorder1.inc” is shown in Figure 5-13, where the circle indicates the position for the guiding codes to be inserted. The diagram of the codes for guidance is shown in Figure 5-14. Only when the file number “\$nf” is greater than 10, ie, only when the student tries to access the circuit connection web page, the feedback on the referred browsing sequence will be given and the students will be advised to visit the file with a hit frequency of 0. The files suggested for visiting are all before the current file in the predefined sequence, ie, their file numbers are all less than the file number of the current file assigned by the “files” table (see Figure 5-12). Feedbacks are sent to the interface by using the PHP3 command “echo”. The viable \$flag (Figure 5-14) is used to indicate the number of those files suggested for browsing. If it is greater than 0, the file “recorder1.inc” will terminate before recording the access of the current file, and the content of the current file will not appear.

After students have begun to use the system, the initial “visiting” table (Table 5-1) is changed (see Table 5-2 as an example). Note that for Group 3 in Table 5-2, only circuit connection for Experiment 2, 3 and 4 haven’t been accessed, while Group 2 has only accessed File 7 and File 1. In this case, if Group 2 tries to access “Cir3.php3” (or the web page of circuit connection for Experiment 3), the feedback shown in Figure 5-15 will be given. Group 3 will receive the feedback shown in Figure 5-16 when it tries to access “Cir5.php3” (or the web page of circuit connection for Experiment 5).

List 5-1: Recorder1.inc

```

<?      //set to php mode
        // Include this file in every page whose access you want
to record

        include('../../monitor/include/connect.inc');

//=====
$file=getenv("REQUEST_URI");
        $filename = basename($file);
$x=strpos($filename, 'x');
if ($x==1){
$f=strpos($file, '/tutorial/exp');
$file=substr($file, $f+9, 15);
}
//=====

        $the_date = date("d/m/Y");
        $logon_time = date("H:i:s");

        $connection = db_connect();
        if($connection > 0)
        {

                $query = "select * from session_details where
(remote_address = '$REMOTE_ADDR') AND (the_date = '$the_date')";
                $result =
mysql_db_query($GLOBALS["database_name"], $query, $connection);
                $row = mysql_num_rows($result);

                if($result == 0)
                {
                        echo "You are not authorised to use this
system";
                                exit();
                }

                $user = mysql_result($result, $row-1, 0);

                $logout_time = mysql_result($result, $row-1, 5);

                if($logout_time != "")
                {
                        echo "You have logged out of the system,
please log back in";
                                exit();
                }

                if ($user != "lec")
                {

```

```

//Guidance*****
$query1 = "select * from files where (filename = '$filename')";
$result1 =
mysql_db_query($GLOBALS["database_name"], $query1, $connection);
        $row1 = mysql_num_rows($result1);

if($row1 > 0){
        $nf=mysql_result($result1, 0, 0);
//-----
        if($nf>10){

                $query1 = "select * from visiting where (ng = $user)";
                $result1 = mysql_db_query($GLOBALS["database_name"],
                $query1, $connection);
                $flag=0;
                $query1 = "select * from files";
                $result =
mysql_db_query($GLOBALS["database_name"], $query1, $connection);

                for ($i = 1; $i < $nf; $i++) {
                        $hits = mysql_result($result1, 0, $i);
                        if($hits == 0){
                                $flag=$flag+1;
                                $file = mysql_result($result, $i-1, 2);

                                echo "You haven't visited the web page
about
                                $file yet. You'd
                                better go there first.<br>";

                                }
                        }
                if ($flag>0)exit();
        }
//-----
        $command = "UPDATE visiting SET f$nf=f$nf+1 where ng = $user ";
mysql_db_query($GLOBALS["database_name"], $command, $connection);
}
//*****
                $Command = "insert into page_accesses
values('$user', '$filename', '$the_date', '$logon_time')";

                if($connection > 0)
                {
mysql_db_query($GLOBALS["database_name"], $Command, $connection);
                }

        }

?>

```

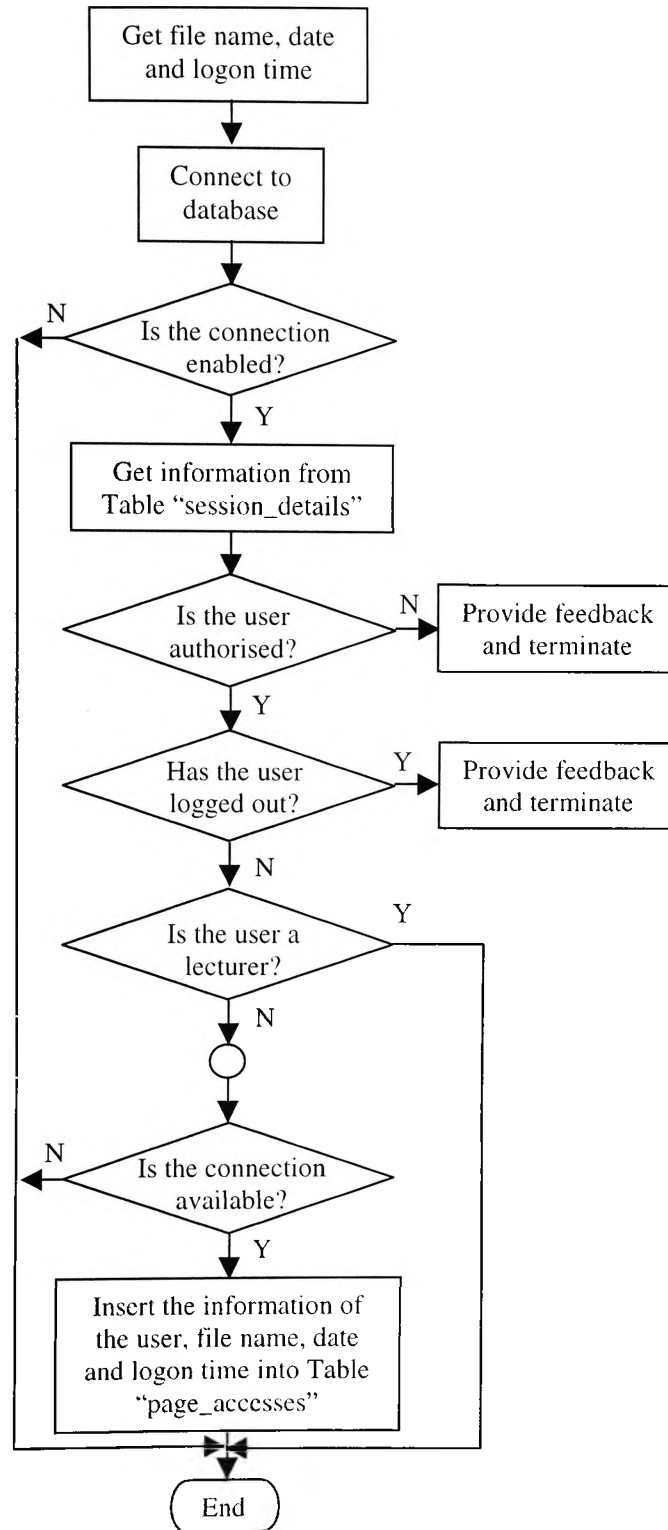


Figure 5-13: Procedure Used in "Recorder1.inc"

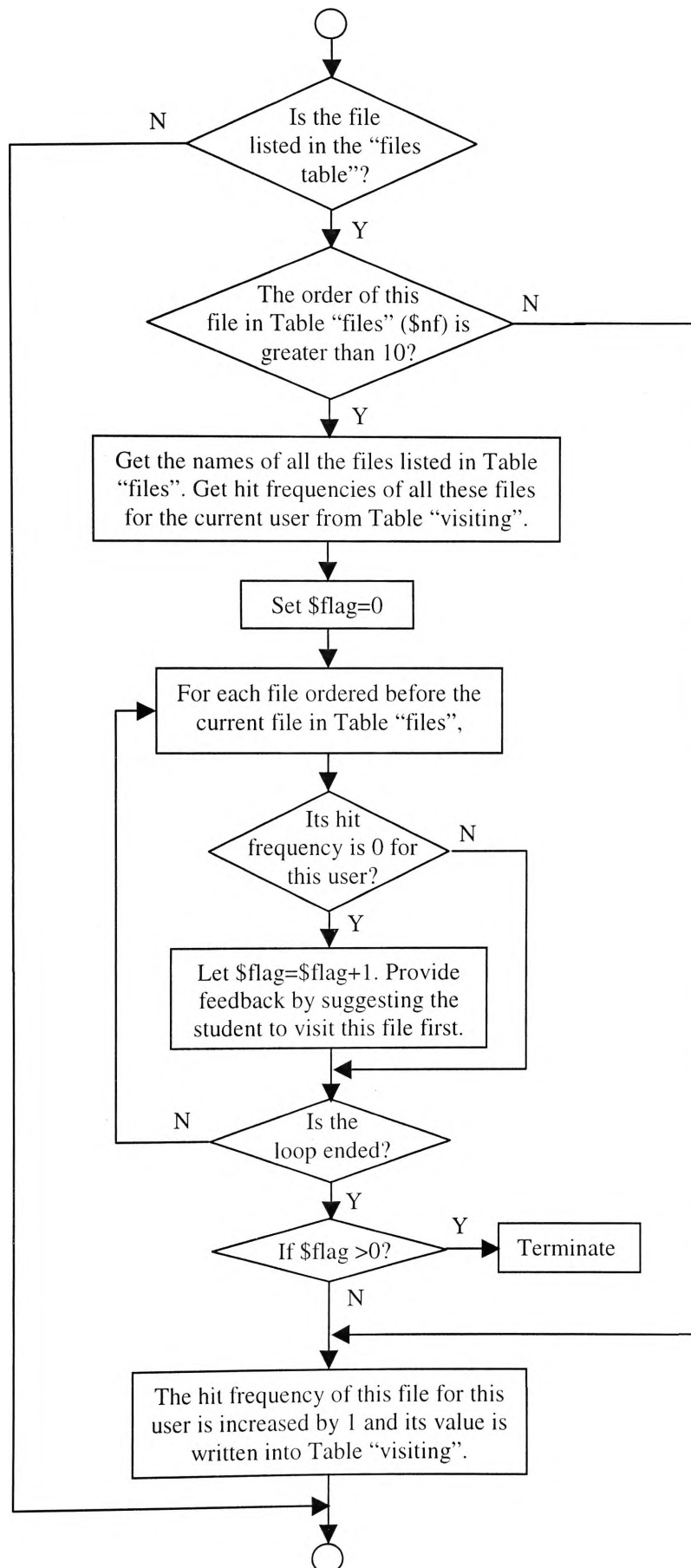


Figure 5-14: Procedure Used in Student Guide

Table 5-2: Contents in the “visiting” Table

Ng	f1	f2	f3	f4	f5	f6	f7	f8	f9	f10	f11	f12	f13	f14	f15
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	2	0	1	0	0	0	0	0	0
3	1	1	4	1	5	1	2	1	1	1	1	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

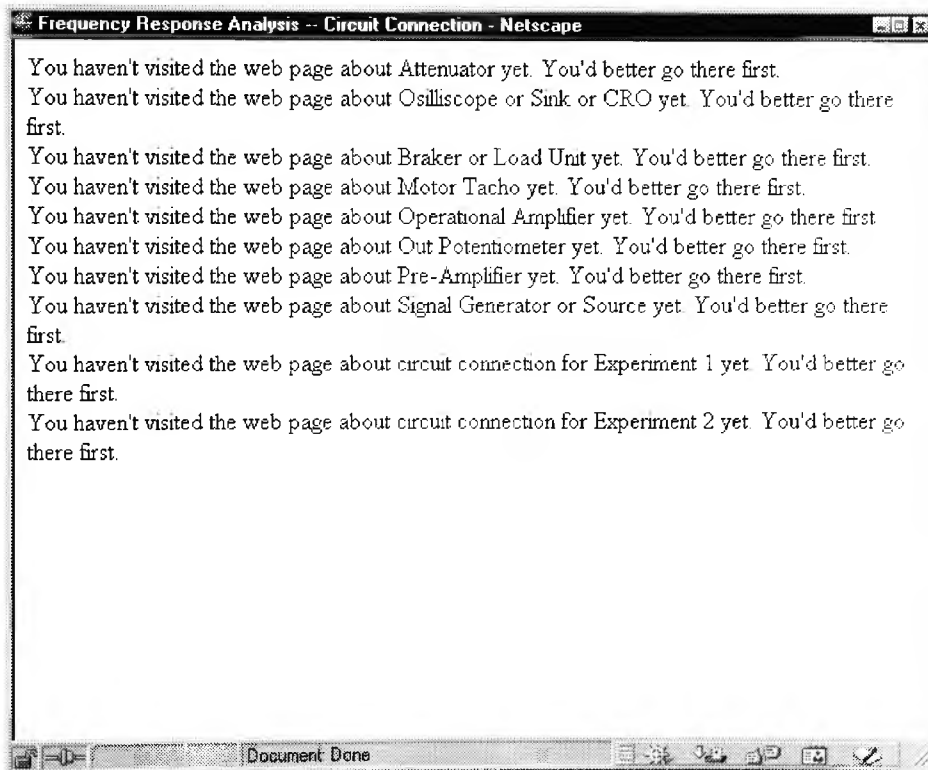


Figure 5-15: Feedback Given to Group 2 for the Access of “Cir3.php3”

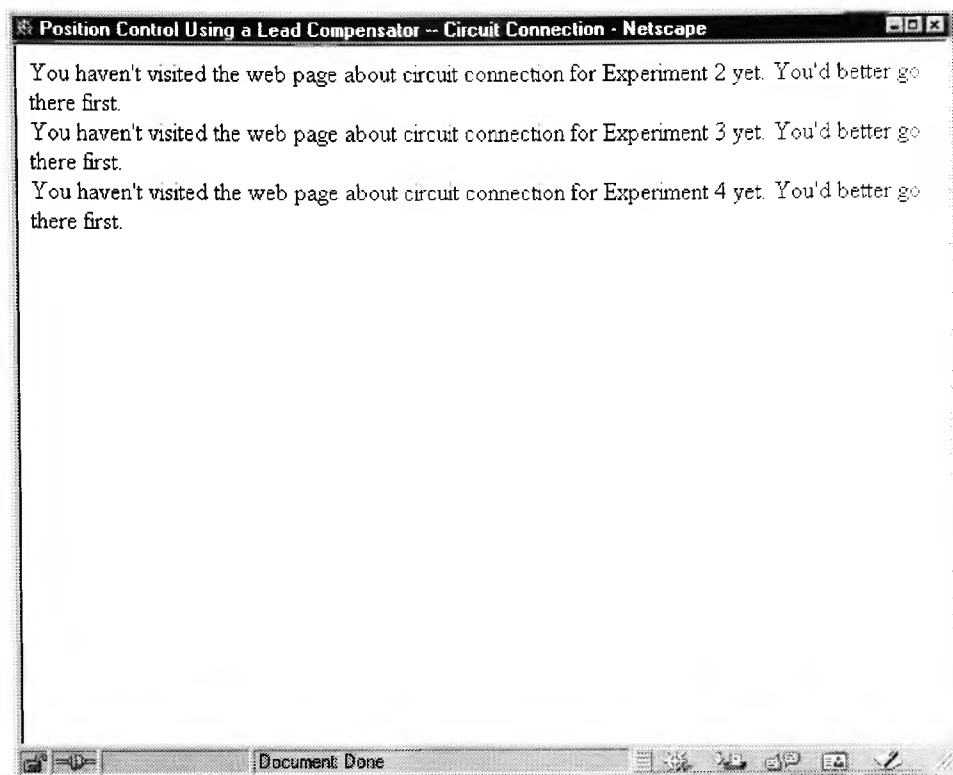


Figure 5-16: Feedback Given to Group 3 for the Access of "Cir5.php3"

5.4 Summary

In this chapter the operation of Student Monitor and Guide have been described and it is shown how they communicate with the Interface Module. Some important programs and their flow charts are also presented.

Chapter 6 : Project Outcomes and Evaluation

6.1 Introduction

The system developed in this work will not only provide instructions and information for the students, but will also record and process students' responses and activities. Frames are used to allow users to navigate through the materials easily. Colour-coded multimedia thumbnails or buttons are used to provide immediate access to multimedia information. Flash movies are embedded in most pages.

Some students and tutors who used the system have confirmed the instructional effectiveness and efficiency of the system. Because of the unique role and purpose of the system, its success is highly dependent on how well it has been developed and whether it works as it was designed to run.

In this chapter the outcomes of the project will be reviewed and the results of its evaluation through students survey will be provided.

6.2 Login Page and Navigation Frame

The user has to log in before he or she can use the package. This requires the user to enter his or her identification code and password (Figure 6-1). The password may be modified if desired. Figure 6-2 shows the interface appearing when a student logs in.

Login Menu

Student 1

Student No.:

Password:

Student 2

Student No.:

Password:

Figure 6-1: Login Page

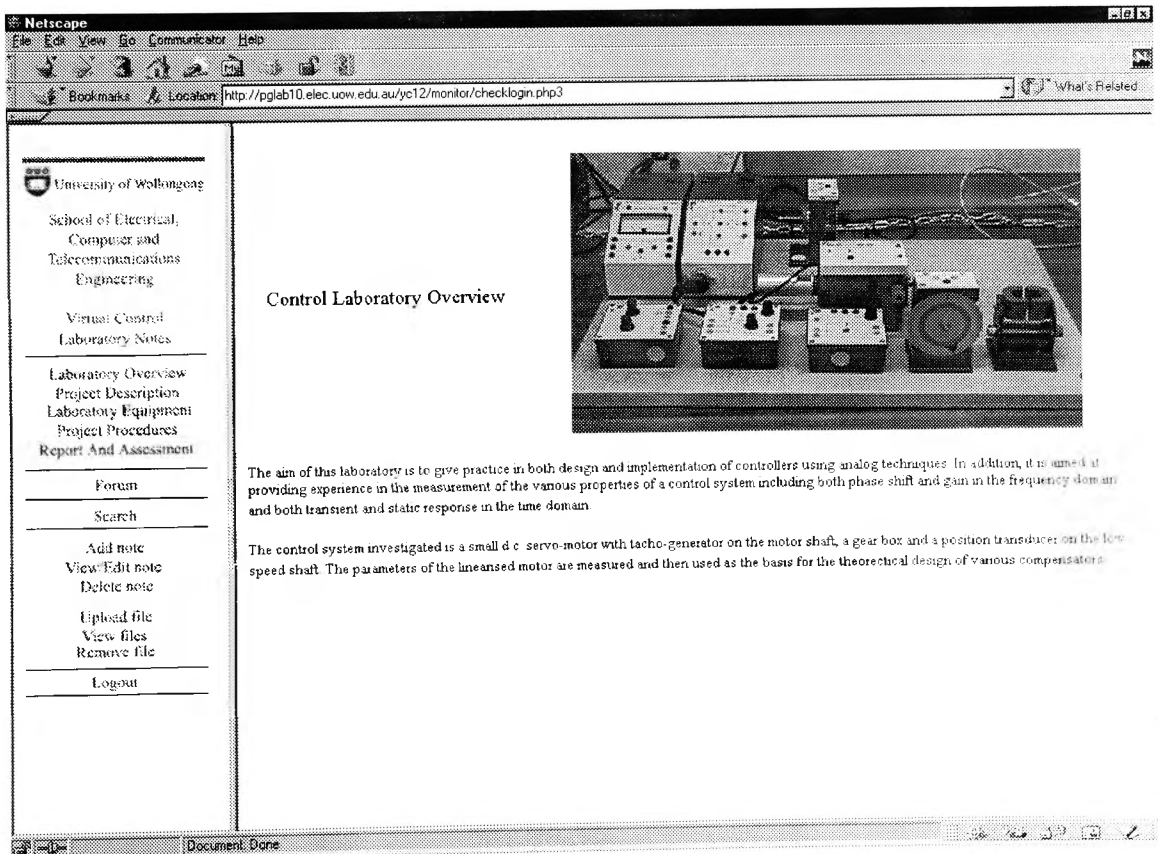


Figure 6-2: User Interface for Student

Two different navigation frames are offered to lecturers and students respectively. The lecturer navigation frame (Figure 6-3) provides access to forum administration and statistics. The web page statistics options allow the lecturer to obtain information on personal details and groups access pages. The student navigation frame (Figure 6-4) hides those functions from the user but adds note-adding and file-uploading functions. The links to instructions are the same for both kinds of navigation frames.

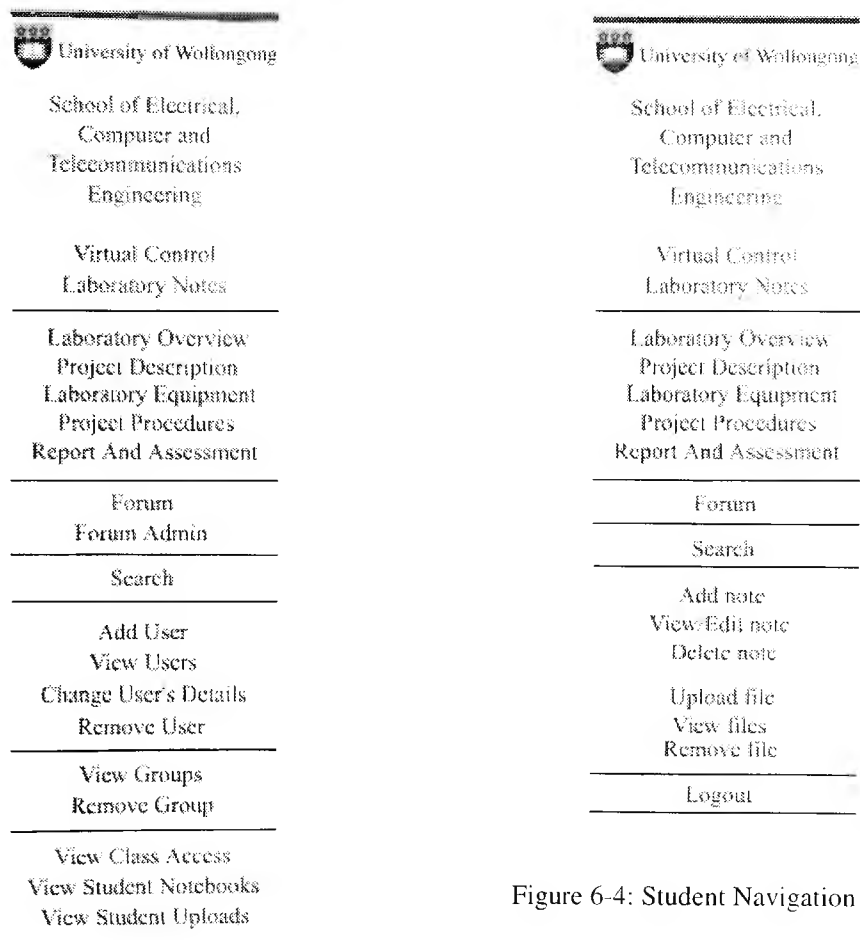


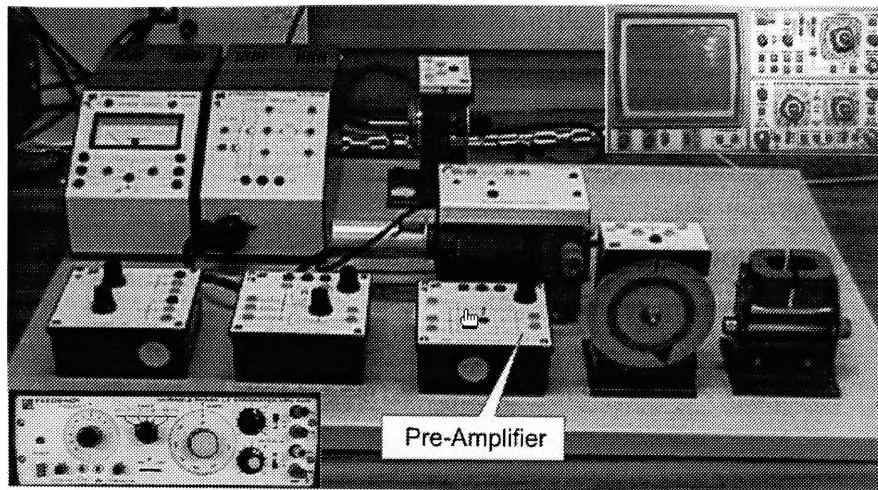
Figure 6-3: Lecturer Navigation Frame

Figure 6-4: Student Navigation Frame

6.3 Laboratory Equipment

6.3.1 Graphical Menu

In the laboratory equipment main page (Figure 6-5), a graphical menu based on an interactive Flash movie is provided. In the menu, all the experimental components are represented in an image, and each component in the image is covered by an invisible button. A text box under the graphical menu explains how the menu can be effectively used. When the mouse arrow rolls over the center of a component, a tag pointing to the component will show the name of the component. The tag will disappear when the mouse arrow rolls out of the component. If the mouse is left clicked when the tag appears, a component pop-up window with only the status bar will appear (Figures 6-6). The status bar remains to show what percentage of the Flash movie has been downloaded in case the server or proxy is too slow. The status bar can also show other information (such as the link or the name of a button when the mouse arrow rolls over the button). The title of this pop-up window will indicate the name of the related component. If another component is clicked, the title and content of the old pop-up window will be overwritten by a new web file containing the information of the new component, which prevents too many pop-up windows to open. The pop-up window can be closed by clicking the “x” bar at the top right corner.



Point to the middle area of each component, you will see its name. Then you can click to see more details of the component in a new window, which can be enlarged as you like.

In the new window of each component, move mouse on sockets, switches, knobs and non-black texts to find more information.

Figure 6-5: Equipment Menu

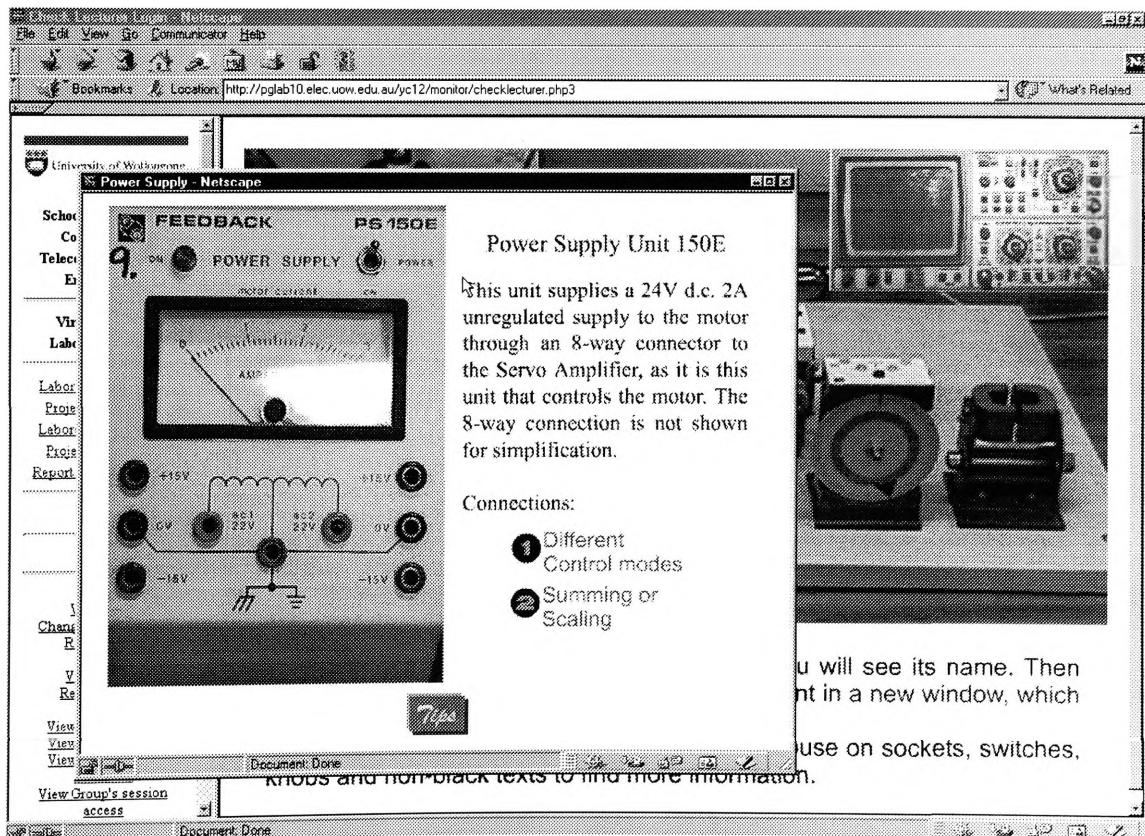


Figure 6-6: Component Pop-up Window

6.3.2 Component Pop-up Window

A Flash movie, containing the image of the components and the explanatory texts, is embedded in the Component pop-up window. At the bottom of the movie, there are one or two blue buttons (depending on the component) which can invoke a Tips pop-up window (Figure 6-7) or a Circuit Diagram pop-up window (Figure 6-8). When the mouse arrow rolls over any socket on the component, a related tag will appear to explain the function or the purpose of the socket (Figure 6-9). When the mouse arrow rolls over a coloured word or phase in the explanatory area, a mark will appear in the picture of the component to show the part mentioned by the coloured word or phase (Figure 6-10). The numbered buttons in the explanatory area will lead to a series of movies illustrating the method of connecting that component to other components in a circuit (Figure 6-11). Clicking the “back” button under the movie will return the screen to the previous movie introducing the component.

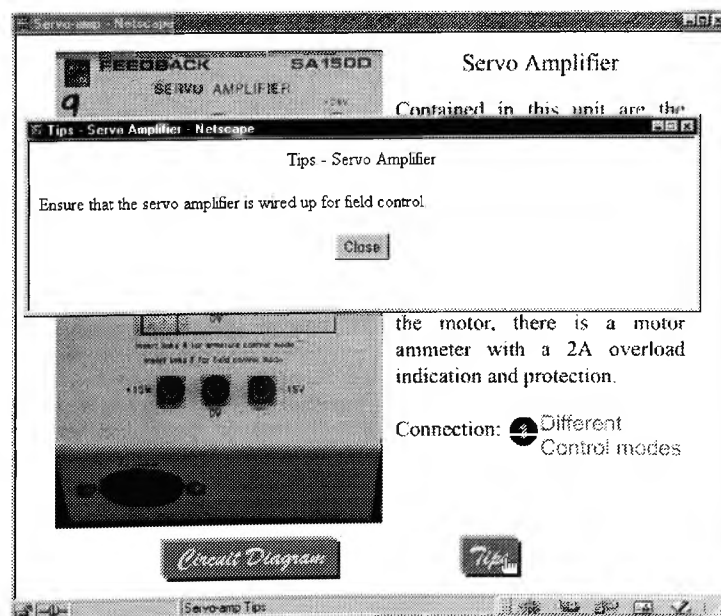


Figure 6-7: Pop-up Window for Tips

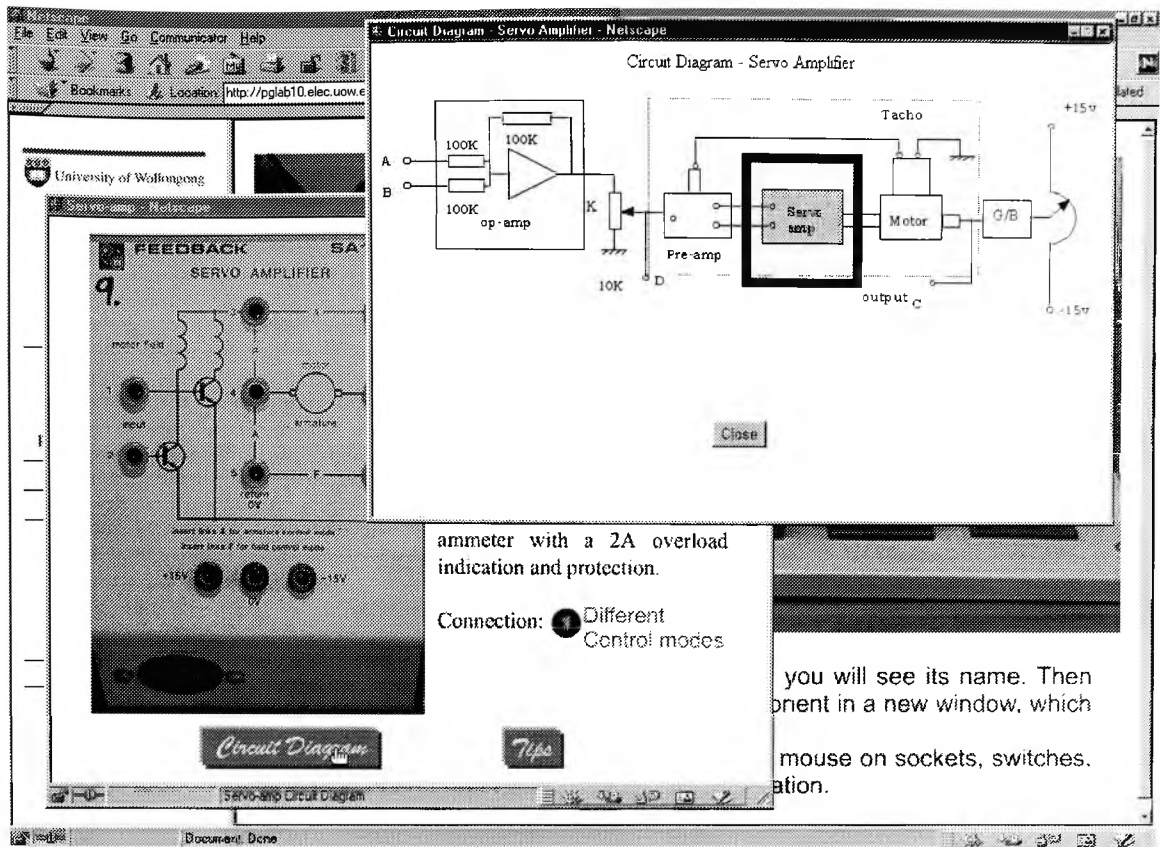


Figure 6-8: Pop-up Window for Circuit Diagram of Servo-Amp

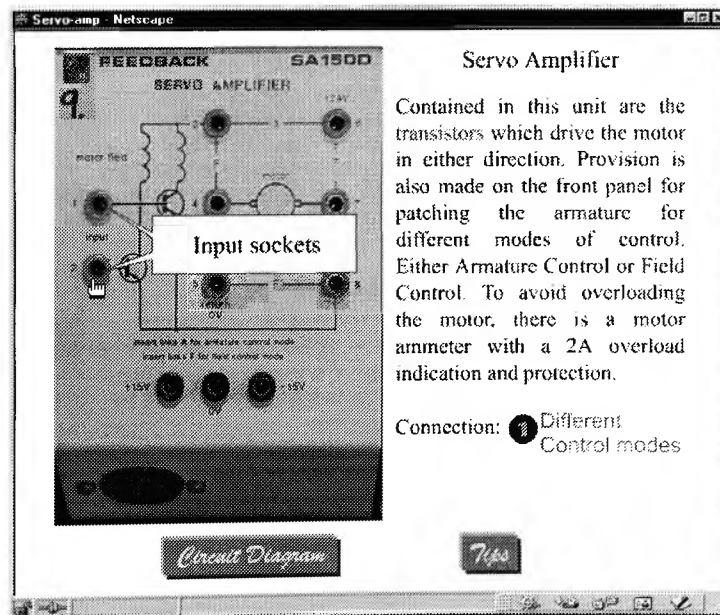


Figure 6-9: Tags for Sockets

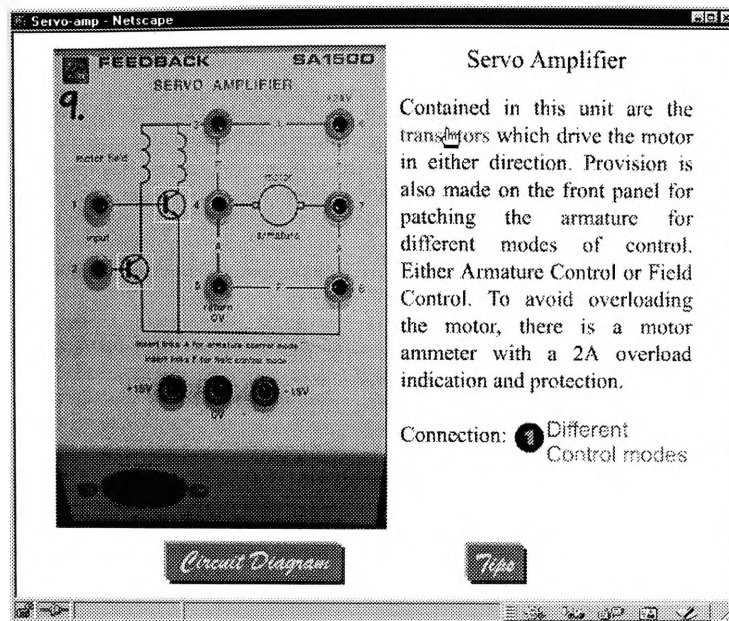


Figure 6-10: Pointing to Coloured Texts

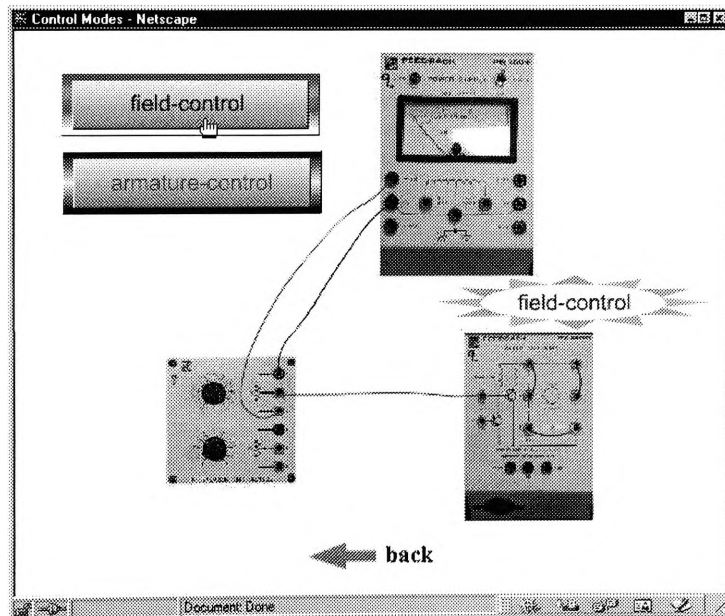


Figure 6-11: Basic Connections

6.4 Project Procedure

6.4.1 Graphical Menu

The Project Procedure main page also contains a graphical menu (Figure 6-12), represented by a Flash movie in which the image together with the title of each

experiment corresponds to a hyperspace node. When the image or the title of an experiment is clicked, a green arrow will point from the experiment image to its sub-titles in the middle of the Flash movie. The sub-titles now include “What you should know” and “Procedures”. They may include more in the future. When each sub-title is clicked, a pop-up window with only the status bar will appear, the content of which relates to the experiment corresponding to the green arrow. The “Troubleshooting” button at the right bottom corner of the movie will lead to the pop-up window for “Troubleshooting”. The pop-up windows for Component, “What you should know”, “Procedures” and “Troubleshooting” all have the same window name “NewWindow”. Hence at a time, only one of such windows can exist, as a new one will overwrite the old one. This makes the navigation simple and straightforward.

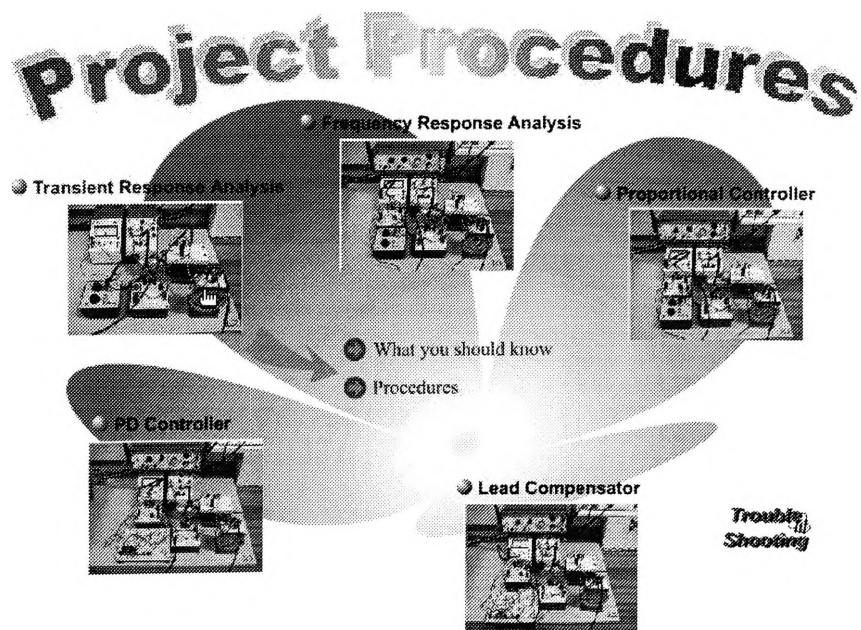


Figure 6-12: Project Procedure Menu

6.4.2 The Pop-Up Window for “What you should know”

In the pop-up window for “What you should know” (Figure 6-13), the experiment title is given above the Flash movie. On the left of the title, there are rewinding, backward and forward buttons for reading the first, the previous and the next page respectively. Under the experiment title there is a sub-title button named “What you should know”. When the mouse arrow rolls over this sub-title button, a pop-up menu appears on its right side (Figure 6-14). The pop-up menu will be hidden when the mouse arrow rolls out the sub-title button. This pop-up menu lists topics in this sub-title and their corresponding page numbers. Clicking a topic in the pop-up menu will change the content below the sub-title button to the content of the corresponding topic with the page number also changed.

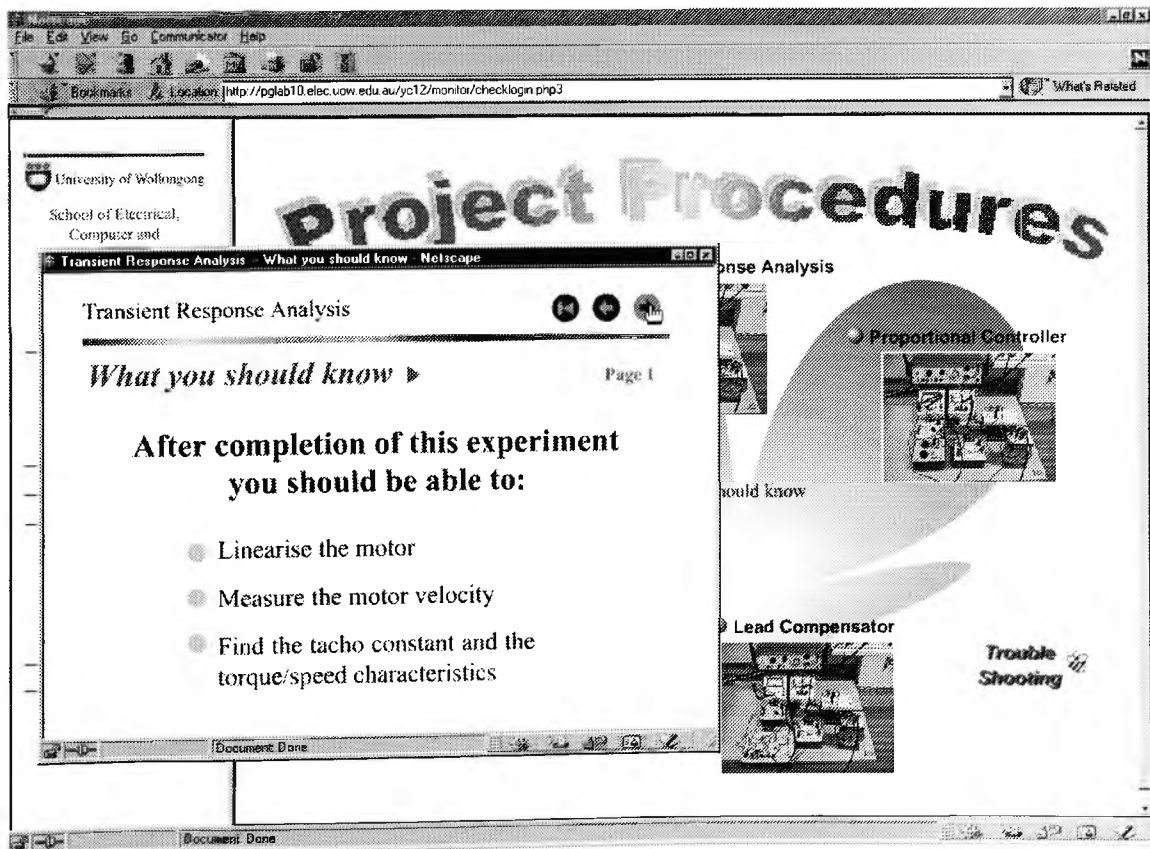


Figure 6-13: Pop-up Window for “What You Should Know”

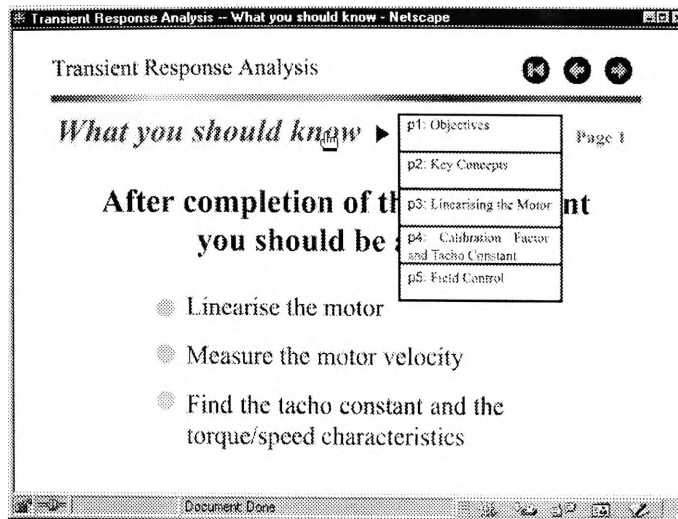


Figure 6-14: Pop-up Menu

6.4.3 The Pop-Up Window for “Procedures”

In the pop-up window for “Procedures” (Figure 6-15), there are three frames. The top right frame is for displaying the RealVideo; and the right bottom frame provides explanation on the video. A video illustrating some views of an experiment will be embedded in the top right frame when the corresponding “Procedures” window of this experiment pops up. All the RealVideos are encoded at the rate of 184kbps for good quality, so communication speeds higher than this rate and RealPlayer version higher than G2 (Real.com, on-line) are required. The videos are synchronised with HTML text files which will appear in the right bottom frame. An HTML text file synchronised with a video may indicate the name of the video or the contents at different stages of the video. In the left frame of the “Procedures” pop-up window, there is a Flash movie similar to the one embedded in the pop-up window for “What you should know” except that video and circuit connection buttons are added inside. When the video buttons at different steps are clicked, the videos corresponding to these steps will appear at the top right frame of the “Procedures” window. When the

circuit connection button is clicked, a new window for selection of components and circuit connection simulation will pop up (Figure 6-16). A student can proceed to circuit connection only when all the components used in a circuit are successfully selected.

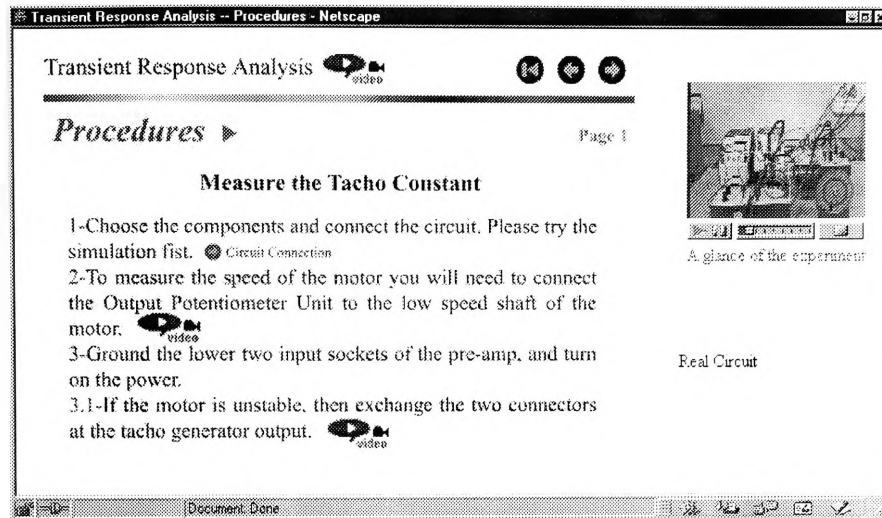


Figure 6-15: Procedure Pop-up Window with RealVideo Embedded

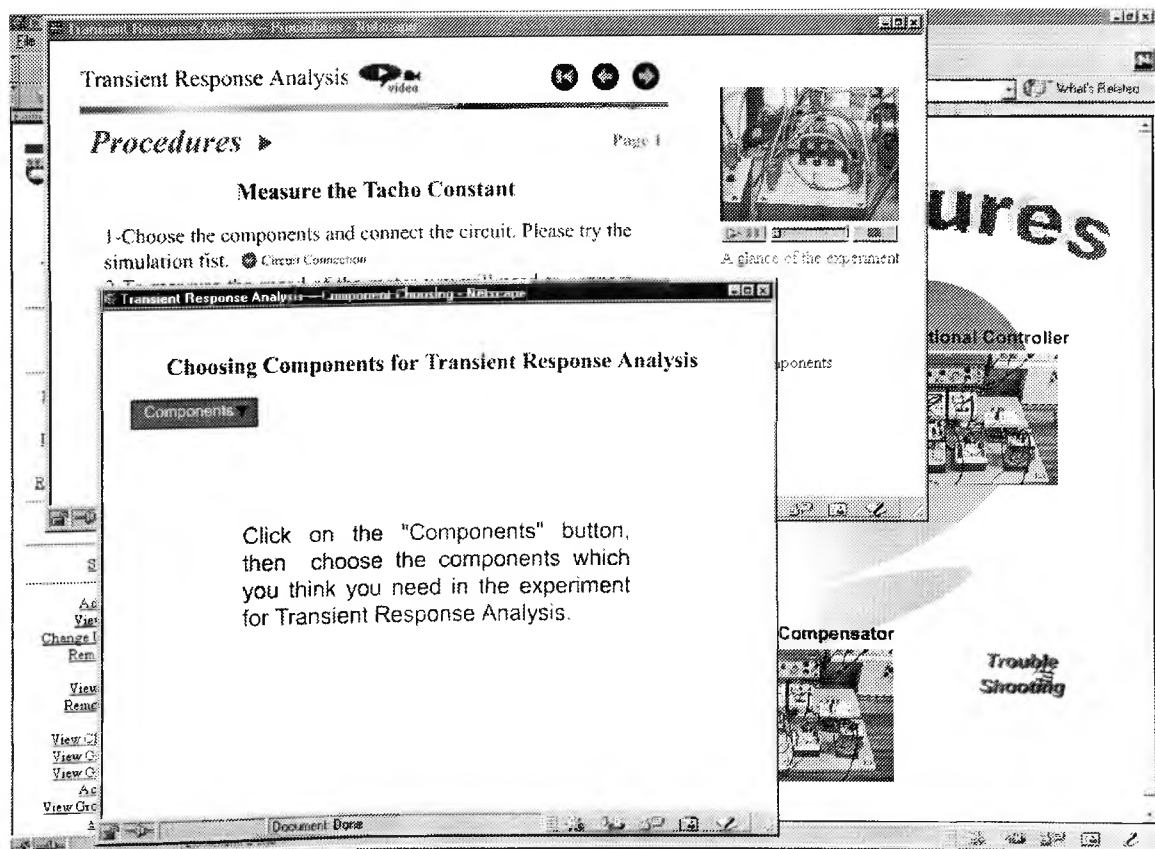


Figure 6-16: Pop-up Window for Component Choosing

6.4.4 Circuit Simulation

In component selection frame, the names of all the components are listed on the left of the frame. When a correct component for a circuit is selected, the image of component appears on the right side of the frame. If a wrong component is selected, a movie clip will appear to inform the student of the error. Some components in some experiments are optional. For example an attenuator can be used in a circuit to protect the circuit at the start up and eventually should be adjusted to not affect the circuit. In these cases, students are informed of such possibility when the mouse arrow rolls over the “Attenuator” button (Figure 6-17).

If the “next” button is clicked before all the necessary components are selected, a movie clip will inform the students that more components are needed for the selected circuit.

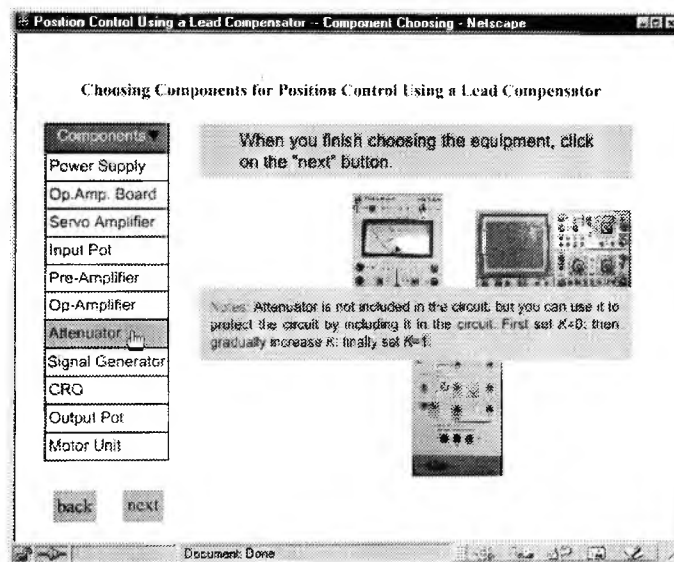


Figure 6-17: Component Choosing Stages

In the circuit connection step, all the components used in the circuit are placed in a Flash movie (Figure 6-18). The student is only required to connect academically

significant connections. Other connections will appear and blink after the student has successfully connected the required wires. Rolling over the “Notes” button highlights the number of wires which should be connected. Rolling over “Tips” button points the input and output signals which should be connected (Figure 6-19). If two terminals of a connection are successively clicked (as mentioned in 4.5.3), a wire indicating this connection will appear. There is no connecting order specified, ie. students are not forced to connect circuit in any order except that they must give two successive clicks as described above to get a connection.

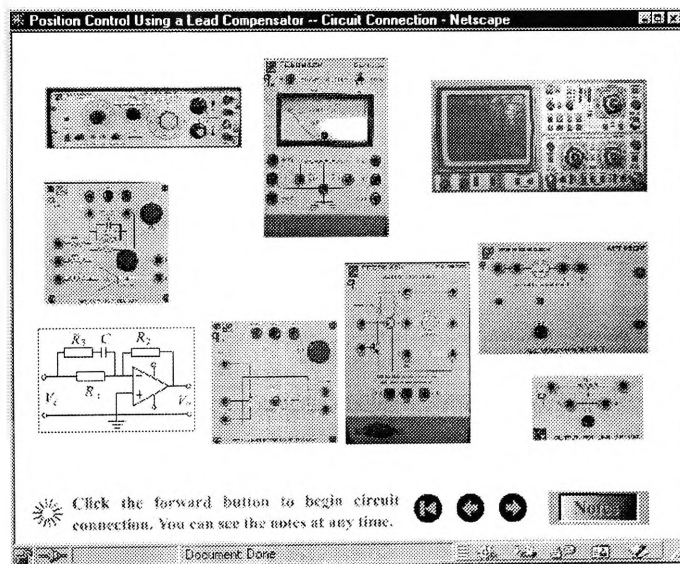


Figure 6-18: Window for Circuit Connection

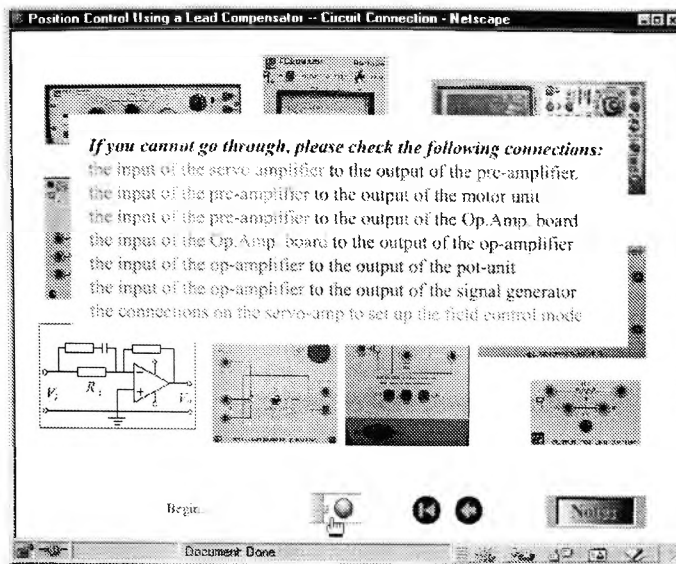


Figure 6-19: Tips for Circuit Connection

6.4.5 Troubleshooting

Troubleshooting button at the right bottom corner of the Project Procedures main menu leads to a pop-up window (Figure 6-20) which provides information on common problems and their solutions. Pictures and animations are used to illustrate the related solutions (Figure 6-21).

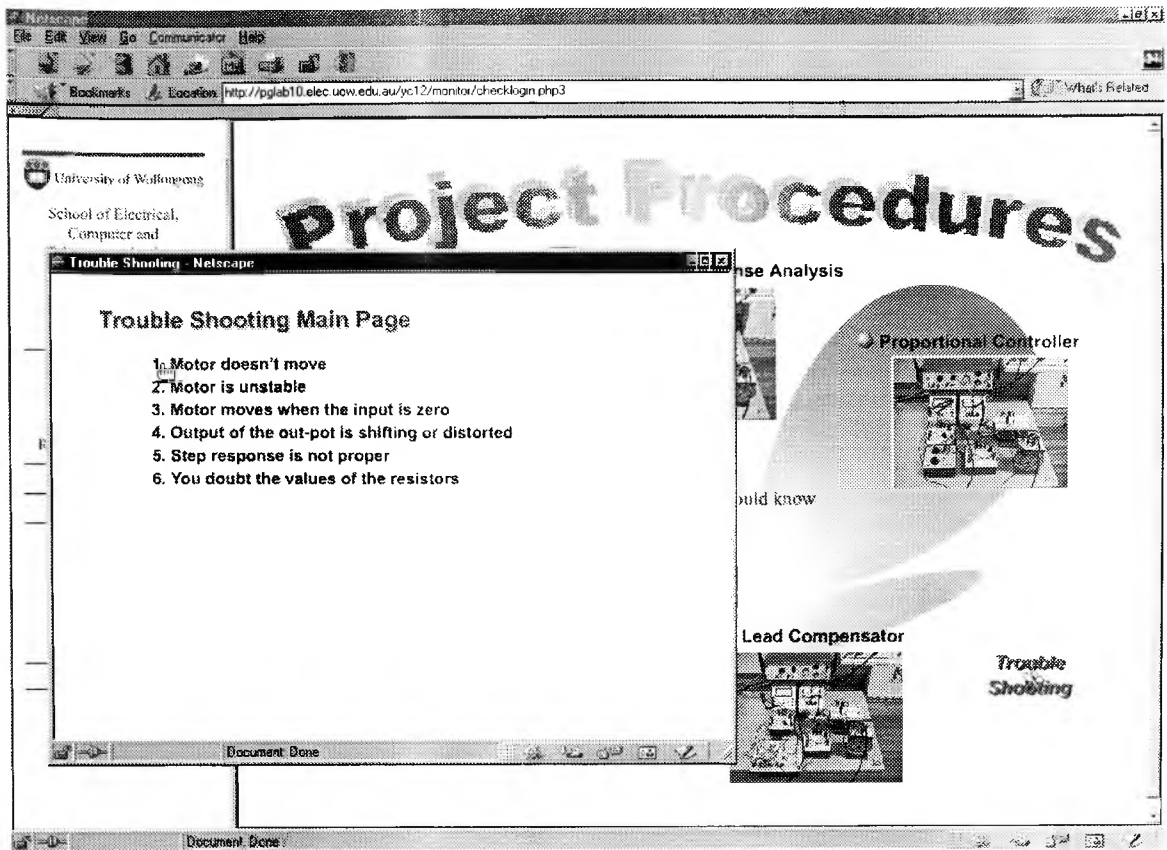


Figure 6-20: Pop-up Window for Troubleshooting

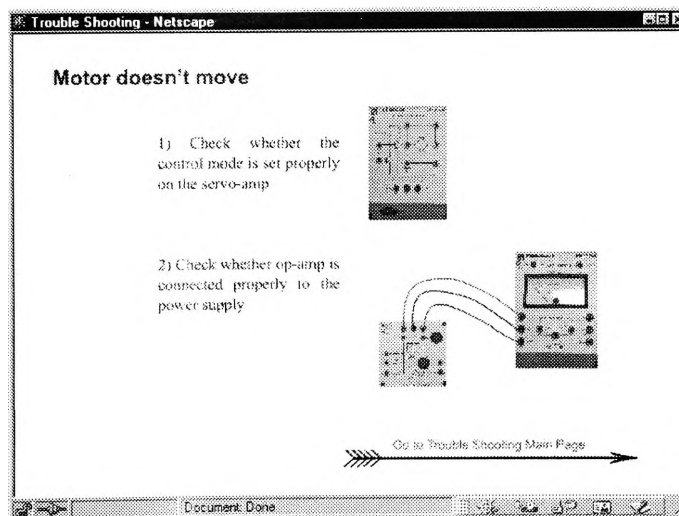


Figure 6-21: Troubleshooting Methods for "Motor doesn't move"

6.5 Other Sections

6.5.1 Project Description, Overview and Assessment

Overview is shown in the right frame of Figure 6-2. The “Assessment” informs the students how their laboratory work is assessed and marked. The “Project Description” describes the content of each experiment and the milestone for each week (Figure 6-22). Some keywords are hypertexts linking to course notes (Figure 6-23) (Naghdy, on-line) or pop-up windows (Figure 6-24) (Rodgers, 1999).

The screenshot shows a Netscape browser window displaying the 'Elec344 - Project Description' page. The browser's address bar shows the URL: <http://pplab10.elec.uow.edu.au/ye12/monitor/checklecturer.php3>. The page content includes a navigation menu on the left and a main table with the following data:

Week	Lecture	Laboratory Milestone
Week 1	Mathematical Modelling of the physical systems including <i>EBlock diagrams</i>, <i>signal flow</i>, state space representation	Introduction to the laboratory and equipment (Odd Weeks)
Week 2	Mathematical Modelling of the physical systems including <i>EBlock diagrams</i>, <i>signal flow</i>, state space representation	Introduction to the laboratory and equipment (Even Weeks)
Week 3	Steady State and Transfer Function - stability issues	Transfer function (Odd Weeks)
Week 4	Block Diagram	Transfer function (Even Weeks)
Week 5	Nyquist Diagram and stability criterion	Frequency Response Analysis - System identification (Odd Weeks)
Week 6	Root Locus Analysis and Design	Frequency Response Analysis - System identification (Even Weeks)
Week 7	Root Locus Analysis and Design	Position Control - proportional control (OW)
Week 8	PI Control	Position Control - proportional control (EW)
Week 9	PI Control Applications	Position Control (PI) (OW)
Week 10	Frequency response design including lead/lag, feedback	Position Control (PI) (EW)
Week 11	Frequency response design including lead/lag, feedback	Lead Design - 1 (OW)

Figure 6-22: Project Description

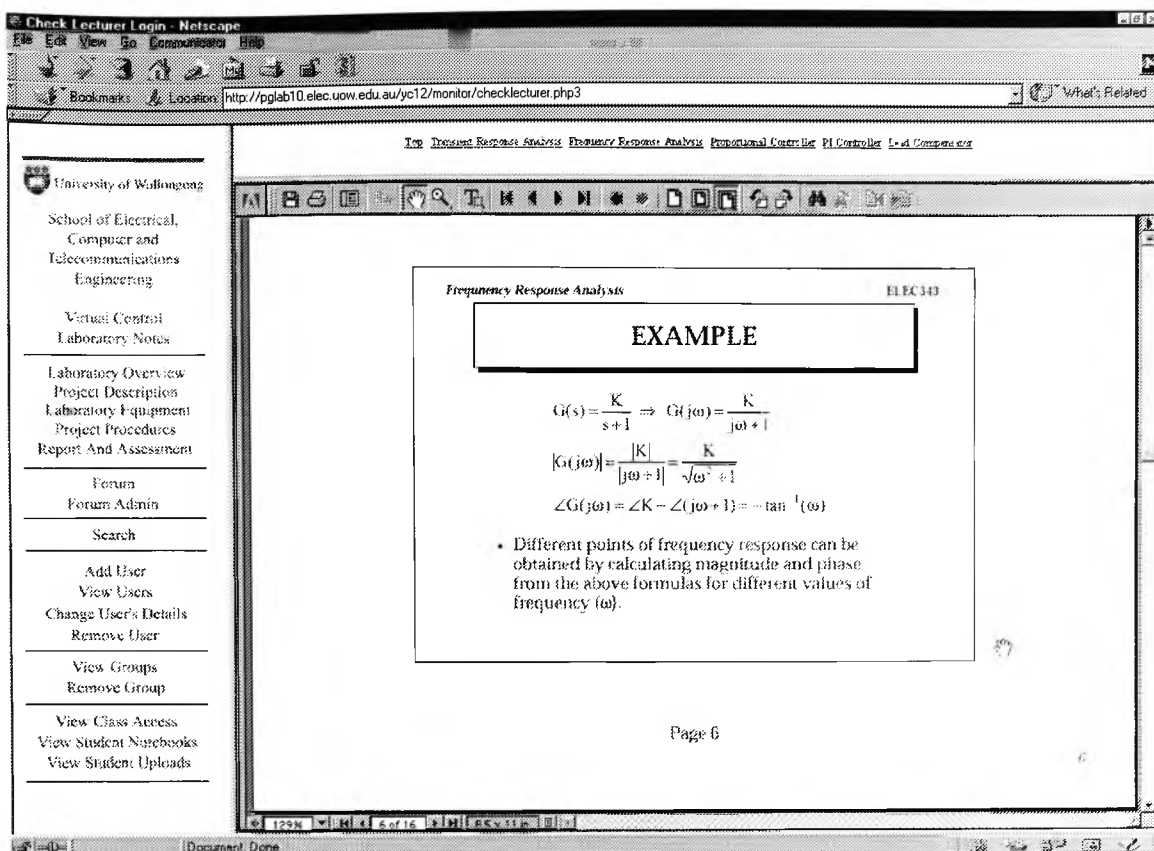


Figure 6-23: PDF Files from Course Notes Website

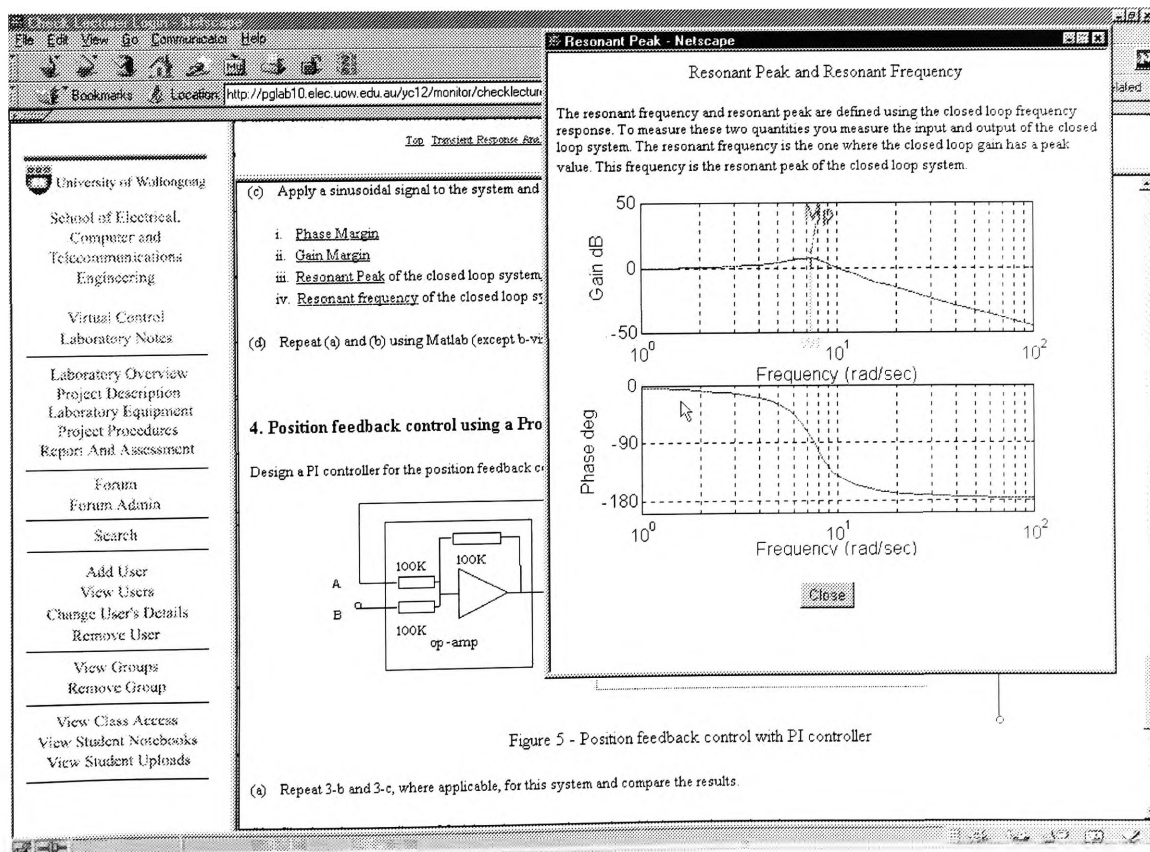


Figure 5 - Position feedback control with PI controller

Figure 6-24: Pop-up Window for Resonant Peak

6.5.2 Facilities: Note Adding, File Uploading, Search Engine and Forum

Students can also add, edit or delete notes and upload or delete files (Rodgers, 1999). They can search information using the Search Engine (Wright, on-line) and ask questions or post messages in the Forum. The forum or message board (Omniboard, on-line) can be viewed page by page (Figure 6-25). Each page is set to display 30 messages. Any message can be followed up. The user can register a name to prevent others using this name. The administrator can delete any message and disallow any IP address.

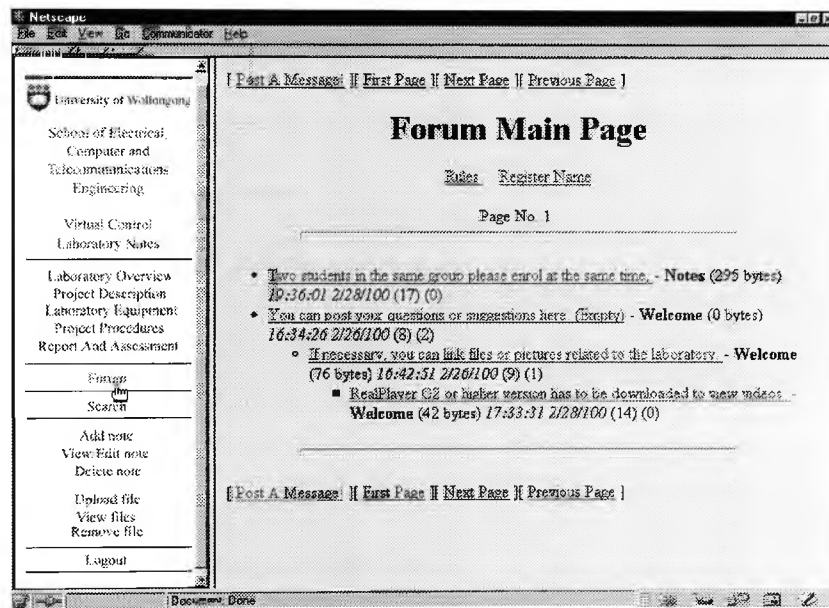


Figure 6-25: Forum

6.6 Evaluation

In order to evaluate the instructional aspects of CALI-II, a student survey was carried out in March 2000. In this survey, 36 students participated. The overall results obtained from the analysis of the received responses indicate that CALI-II has mostly

achieved the instructional objectives set for it during its design and development. It has been well received by the students, accepted as a delivery system, has assisted the students to perform better in the laboratory and to develop a better understanding of the topic. The survey form is provided in List 6-1 and the results are analysed in the following sections. Figure 6-26 illustrates the responses received for questions 1-7.

List 6-1: Elec344 Student Survey

	Strongly Agree	Agree	Not Sure	Disagree	Strongly Disagree
	1	2	3	4	5
Q1: Did you find the website easy to use?					
Q2: Did you find the website informative?					
Q3: Did you find the structure of this interface easy to navigate?					
Q4: Was the instruction provided on the Web clear?					
Q5: Was the laboratory instruction easy to follow?					
Q6: Was the information given regarding the laboratory equipment useful?					
Q7: Did you find the tips useful?					
Q8: How good did the website compare to conventional lab notes on paper?					
Q9: How much have you learned from the laboratory?					
Q10: Are there any areas of the website which you think could be improved?					
Q11: Can you think of additional features that could be added to the website?					

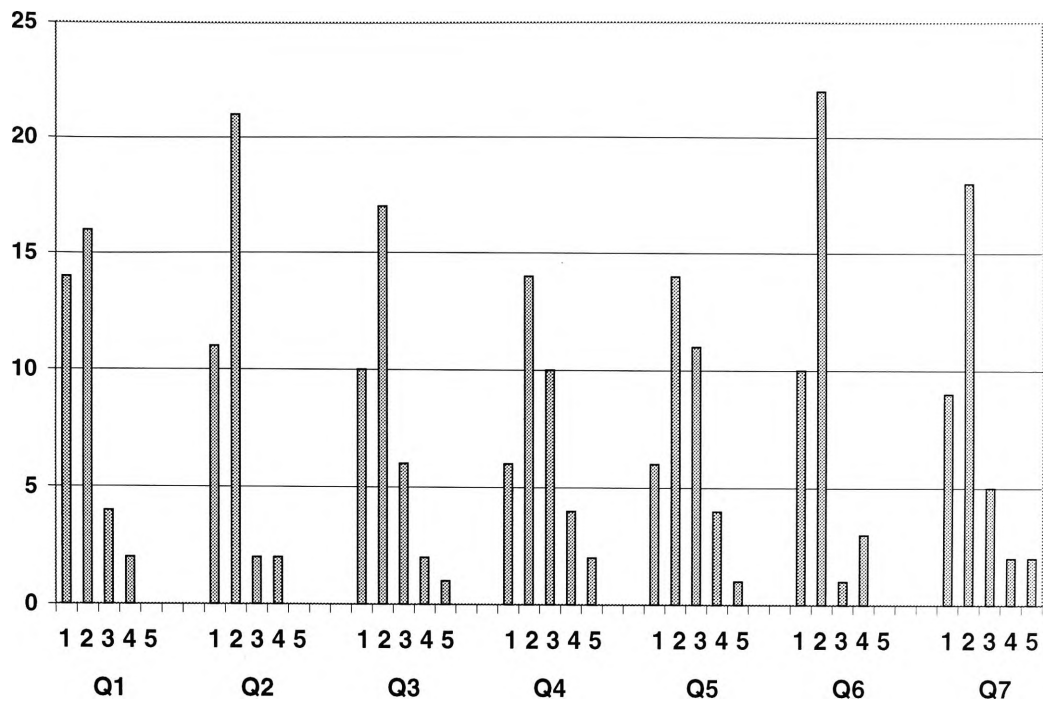


Figure 6-26: Column Chart of Student Survey Results for Q1~Q7

Q1: Did you find the website easy to use?

This question attempts to evaluate the ease of use of the various aspects of the package including accessing a page, navigation, obtaining the required information such as the key concepts or procedures of an experiment. 39% of the students strongly agree that the system is easy to use (Figure 6-27) and more than 40% only agree with the statement. There is no student strongly disagreeing with the statement. The system was introduced to the students for the first time and some of the disagreements could be due to an improperly chosen parameter in the “user” table definition which prevented some of the students to log into the system in the first week or so. Now the problem is fixed.

Overall, the response clearly illustrates that existing generic computer skills in a student is sufficient to utilise the package and no extra learning is required.

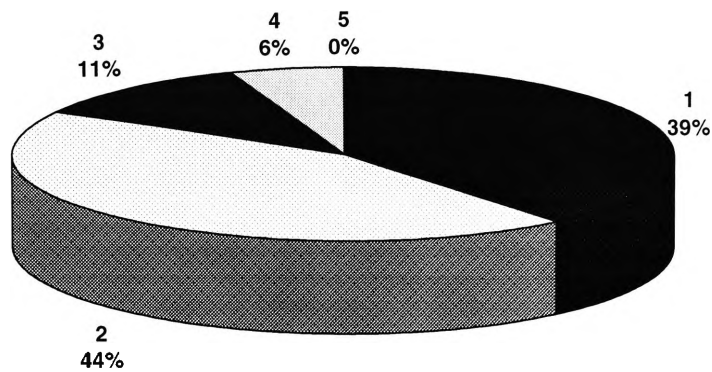


Figure 6-27: Pie Chart for Q1

Q2: Did you find the website informative?

This question is intended to evaluate whether CALI-II provides the students with the information they need to carry out their tasks. 57% of the students agree that the system is informative (Figure 6-28) and more than 30% strongly agree. There is no student who strongly disagrees with the statement. There are only 6% of the students who disagree and the same percentage cannot decide. More study is needed to identify what other information is required by the latter group of the students.

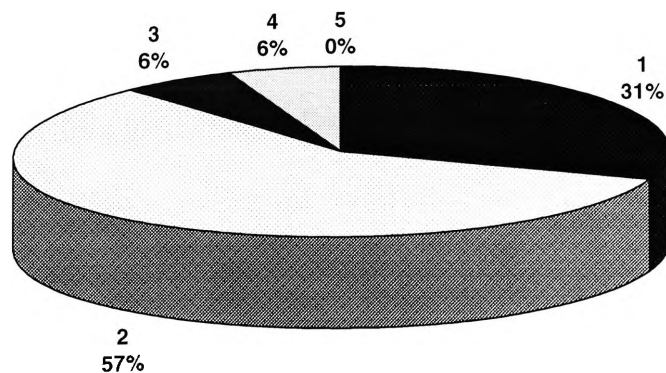


Figure 6-28: Pie Chart for Q2

Q3: Did you find the structure of this interface easy to navigate?

This question is concerned with the navigation structure of CALI-II and has some overlap with question 1. There are 28% of the students who agree the system is easy to navigate (Figure 6-29) while 46% who strongly agree. There are 3% of the students who strongly disagree with the statement, and 6% only disagree. There are also 17% of the students who cannot make up their mind. This is a little strange since by the time of the survey, the students had used the system for two laboratory sessions (6 hours). Further study will be required to identify the nature of the concern of such students and amend the system accordingly.

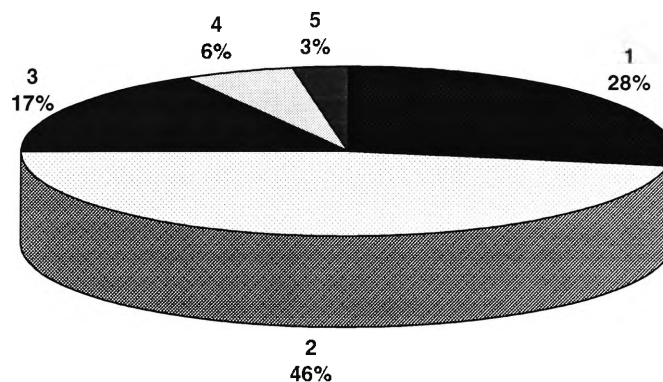


Figure 6-29: Pie Chart for Q3

Q4: Was the instruction provided on the Web clear?

The focus of this question is on the clarity of the instruction in CALI-II. There are 38% the students who agree the system provides a clear instruction (Figure 6-30), while 17% strongly agree. There are 6% of the students who strongly disagree with the statement, 11% disagree, and 28% who cannot make up their mind. The high-percentage of the students who are unsure of their responses also need more

investigation. This could be due to the short period that the students have been using the system. The complete laboratory takes 6 sessions (18 hours).

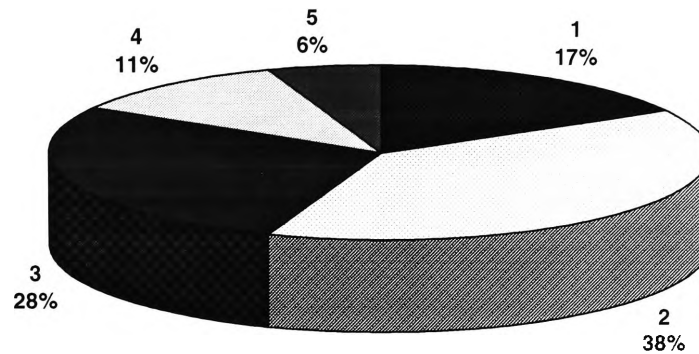


Figure 6-30: Pie Chart for Q4

Q5: Was the laboratory instruction easy to follow?

Question 5 is also concerned with the instruction but tries to identify how easily students could implement the instruction. There are 17% of the students who strongly agree with the statement (Figure 6-31), 38% students who just agree and 31% unsure. There are 3% of the students who strongly disagree with the statement, and 11% of the students disagree. The result obtained from this question is very similar to the previous question which may suggest that the majority of the students did not understand the question and gave the same response as question 4.

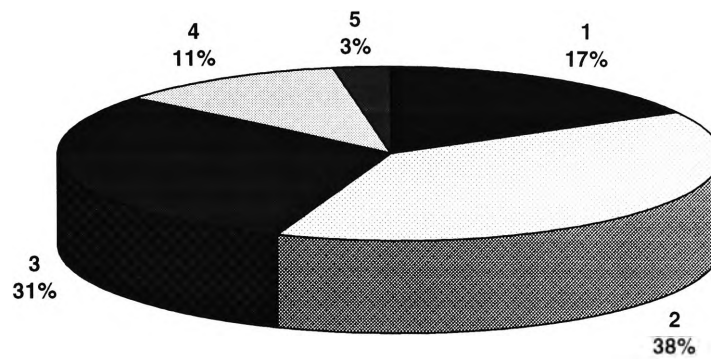


Figure 6-31: Pie Chart for Q5

Q6: Was the information given regarding the laboratory equipment useful?

This question evaluates the instruction provided on the laboratory equipment. There are 28% of the students who strongly believe the instruction is useful, while 61% only agree. There is no student who strongly disagrees with the statement, while 8% disagree and 3% are unsure.

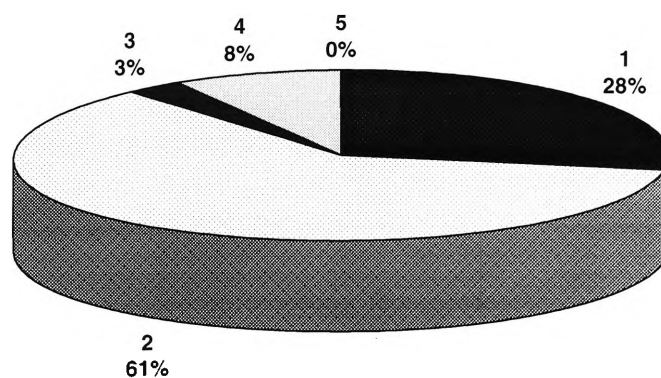


Figure 6-32: Pie Chart for Q6

Q7: Did you find the tips useful?

Question 7 evaluates the usefulness of the tips. There are 25% of the students who strongly agree that the tips are useful, 49% only agree, 6% strongly disagree, 6% disagree and 14% are unsure.

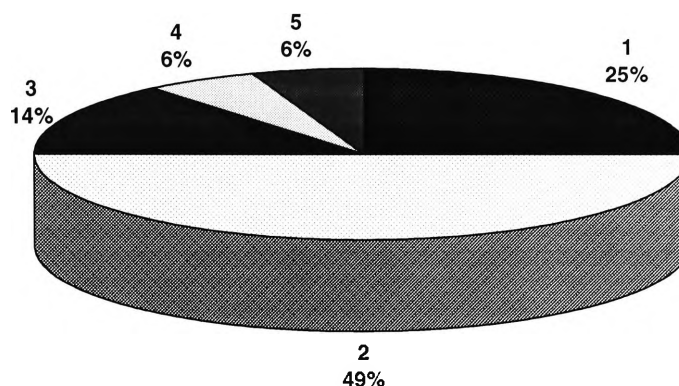


Figure 6-33: Pie Chart for Q7

O8: Comparison with Conventional Laboratory Instruction

Question 8 evaluates the effectiveness of CALI-II compared to conventional laboratory instruction and paper-based laboratory notes. The majority of the students commented that

- CALI-II was much more attractive, definitely more illustrative and interactive, and much better than the paper-based conventional laboratory notes.
- The information provided by the website was clearer and easier to understand.
- The use of real-life pictures and videos was quite effective.
- They liked open access to the website out of the laboratory session for preparation and further work.
- It was fun to use CALI-II and perform the simulations.
- They used the system very frequently.

Q9: Depth of Learning

The purpose of this question is to assess how much students have learned from the laboratory sessions using CALI-II. As mentioned before, the surveyed students were only two sessions through their 6-session laboratory work. In spite of that, the majority of the students felt that they had learned a great deal and more than what they expected. They also expressed enthusiasm to continue the laboratory, to learn more and to complete their work.

Q10: Improvements

In this question students are asked what can be changed in CALI-II to make it more effective and useful. The main complain was that they could not access the page from outside of the University. This was a restriction imposed by the University Administration, not as the result of the operation of the package.

In addition, some students also did not know how to print information in a pop-up window. The instruction is now posted on the Forum.

Some students expected to have more details on the experiment and the operation of the equipment. The details associated with the equipment may be added later but more information on the analysis will reduce the active role of the students in their learning process as they are expected to carry out such analysis themselves.

Q11: Additional Features

In this question students are asked what other features can be added to CALI-II to enhance it. The majority of the students commented that the package already had sufficient features. Some felt that quizzes and exercises might be good additions to the site.

Lecturer Access

A number of the laboratory tutors were asked to evaluate the Lecturer Access of the system. The overall response received was very positive. It was commented that the system was fairly consistent, effective and versatile. Some thought that allowing the students to add notes and upload reports into the system was a good idea. They also liked the idea of sequencing the on-line notes.

6.7 Summary

The system outcomes and evaluation were presented in this chapter. The functions of different aspects of the developed CALI-II were described and it was clearly shown that the objectives set for the project have been achieved. In addition, the result of an evaluation of the package carried out based on the student-surveys was also presented.

Chapter 7 : Conclusion

7.1 Introduction

This chapter will summarise the findings of the project, will draw some conclusions from such findings and will propose new approaches and further work to improve the effectiveness of CALI-II.

The findings and conclusions drawn from the project can be divided into two groups. The first group consists of specific outcomes which relate directly to the particular application of the approach in Control Laboratory and the benefits it has produced. The second group of the outcomes are generic. They represent the features of the approach which can be realised in other applications. Both types of the outcomes will be studied in this chapter.

7.2 Specific Outcomes

The development of CALI-II has significantly improved the quality of the work carried out by the students in the Control Systems laboratory and has enhanced their efficiency. This is due to specific and unique features of the package:

Ease of Use and Modification

The content of the project is delivered by a Web-browser. Frames, pop-up windows and pop-up menus are employed to make navigation clear and easy to use. Due to the simplicity of the interface, students can become familiar with the package through self-exploration, with little help from the instructor. Similarly, the system is very easy for the instructor to set up, read various databases and track the students' activities.

Rich Multimedia

The textual representation of information is augmented by rich multimedia, including pictures, animations and videos which are not possible in printed media. This has proved quite effective to attract students' attention, motivate them to work efficiently, and provide them with applications and examples that supplement the theoretical information. In addition different media formats can cater for students' various preferences.

Convenient and Quick Access

Every student can access the same knowledge-base at any time without waiting for the scheduled laboratory sessions. Information can be accessed quickly without long download time. Flash is used to design vector-based animation and RealSystem is used to create synchronised RealVideo files and stream them through Web. These authoring tools, especially developed for WWW, can create small-sized multimedia files suitable for the limited bandwidth of WWW. In addition, the fast server-side scripting language — PHP3 and a very fast database server — MySQL have been used in the development of the package.

Student Monitor

The Student Monitor records students' browsing activities and provides important information for the intelligent system to guide students. The lecturer or instructor can also use this information to analyse students' activities and attendance. Using new techniques on Common Gateway Interface, Database and Fill-Out Forms, the monitor is established with almost no effect on the browsing speed.

Guidance and Presentation Control

Presentation control is employed to guide the students in their navigation and keep them on the right track to use the pages effectively and in the right order.

Circuit Simulation

A circuit connection simulation enables the students to develop the necessary knowledge and skills for setting up a specific circuit before they actually wire up the experimental rig.

Iconic Interaction

Large graphical displays or icons are created as buttons or hypernodes to allow students to participate in the domain directly. This encourages the students to obtain new information within the system without getting frustrated and bored.

Transparency and Consistency of the Interface

The interface of the system is transparent, ie, the domain semantics is visible from the interface. In addition, a consistent layout for displaying information of the same type throughout the package assists the students to get used to the environment quickly and navigate effectively.

7.3 Generic Outcomes

CALI-II has set new standards and introduced new approaches in Computer Assisted Laboratory Instruction. The methodology developed in this work is quite unique while at the same very simple and clear. Here are some of the generic outcomes identified for the project:

- (a) This work has clearly demonstrated that Computer-Assisted Instruction is a viable concept and can be achieved. The methodology developed in the work in terms of its overall structure, design of the frames, instructional design, Student Model, etc can be easily realised for another application by changing the content.

- (b) It has been shown that Web-Based Computer Assisted Instruction Systems are not inferior to standalone PC based applications developed using other authoring tools. The tools employed in this work for developing interactivity have produced multimedia outcomes comparable with, if not superior to, non-Web tools such as Authorware and Quest.
- (c) Comparison of CALI-I and CALI-II shows that Web-based applications are superior to standalone applications in terms of accessibility and usability.
- (d) The shortcomings of the conventional laboratory instruction identified in this work and the solutions devised seem to be true as the application of CALI-II in the laboratory has significantly addressed such problems.
- (e) Although further work is required to evaluate the significance and impact of the package, its initial employment clearly has produced a better performance by the students and a deeper understanding of the activities they perform.

7.4 Further Improvement

The effectiveness of CALI-II in instruction can be improved further by introducing new facilities and tools for it.

Further Evaluation

The package should be evaluated more systematically and over a longer period to identify all the inherent bugs and also clearly to identify its effectiveness in enhancing the performance of the students and the depth of their learning.

On-line Tests

In addition to the current method of evaluating the performance of the students by tracing their activities, further active steps in assessing the students can be made by

introducing on line tests such as multiple-choice questions. The tests can be also used as a method to encourage the students to review the course materials by providing hints or hyperlinks to on-line materials associated to each question in case the students need assistance to answer a particular question.

Diagnostics Checks

The results obtained by the students for different parts of the experiment can be compared with reference values kept in a database to determine how true and correct the measurements are carried out by the students. Depending on the compared values, various remedial actions and procedures can be suggested to the students to improve their results.

Indexing

Indexing can be introduced to index the contents of CALI-II by frames, topics, keywords or concepts to make instructional materials more flexible in the hypermedia area since the indexed sets are not related to any pre-scribed order of presentation. Indexing concepts of each unit (or topic) will allow the system to know what is presented in each unit.

Appendix A : Installation of CALI-II

In this appendix, issues related to the installation of CALI-II including the required environment on the server to run the package successfully are described.

The CD containing the package consists of a large number of files structured in different directories, as shown in Figure A-1, should be uploaded into the base directory on the server. According to this structure, the cgi (Perl) files for Forum (Omniboard, on-line) and Search Engine (Wright, on-line) are uploaded into the subdirectory \yc in \cgi-bin (Figure A-2), and the RealVideo (.rm) files are uploaded into \Realserver\content (Figure A-3).

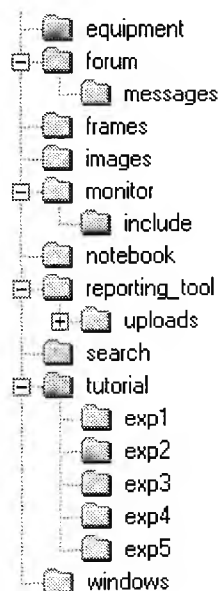


Figure A-1: Directories Structure of CALI-II
Which Should Be Set Up at the Base of the
Server.



Figure A-2: Directory Holding the .cgi Files



Figure A-3: Directory Holding the .rm Files

A dummy file named “index.html” should be placed in the “\monitor” and “\forum” directories to prevent unauthorised people to access these directories.

The access permissions associated with the following directories or files should be changed according to Table A-1. The Unix or Linux command for changing the access permission is “chmod”. If the system is set up for a different website, then accordingly the associated codes in a number of forum and search engine files should be changed. Details of such modifications are provided in the “Readme” files in \forum and \search sub-directories.

Table A-1: Access Permissions for different files and directories

Directories or files	Permissions	Additional information	Directory
\uploads	rw-rw-rw-rw-		All in the base directory
\forum	rw-rw-rw-rw-		
\messages	rw-rw-rw-rw-		
mainboard.html	rw-rw-rw-	initial value = 1	All in \forum
serial.txt	rw-rw-rw-		
xxxx_bywhom.txt	rw-rw-rw-	Disable any IP.	
xxxx_dup.txt	rw-rw-rw-		
xxxx_badguy.txt	rw-rw-rw-		
xxxx_regpwd.txt	rw-rw-rw-		
xxxx_admnpwd.txt	rw-rw-rw-		
addnew.html	Permissions don't need change	Codes may need change if the system is set up for a different website	
register.html			
yyyy_admboard.html			
search.cgi	rw-r-xr-x	Codes may need change if the system is set up for a different website	All in \cgi-bin\yc
mainpage.cgi	rw-r-xr-x		
mainboard.cgi	rw-r-xr-x		
register.cgi	rw-r-xr-x		
yyyy_admboard.cgi	rw-r-xr-x		
yyyy_admnquick.cgi	rw-r-xr-x		

A database named “elec344” should be created in MySQL in the server. The database will require a number of tables with the following names: notebook, page_accesses, reports, session_details, user, files and visiting. MySQL table definitions for “notebook”, “page_accesses”, “reports”, “session_details” and “user” tables can be found in (Rodgers, 1998). The MySQL commands for creating Tables “files” and “visiting” are listed in Lists A-1 and A-2 respectively.

List A-1: MySQL codes for creating Table "files"

```

use elec344;
Drop table if exists files;

CREATE TABLE files (
nf tinyint(4) DEFAULT '0' NOT NULL AUTO_INCREMENT,
PRIMARY KEY (nf),
filename varchar(25),
forwhat varchar(35)
);
INSERT INTO files (filename,forwhat) VALUES
('attenu.php3','Attenuator');
INSERT INTO files (filename,forwhat) VALUES
('cro.php3','Oscilloscope or Sink or CRO');
INSERT INTO files (filename,forwhat) VALUES
('load.php3','Braker or Load Unit');
INSERT INTO files (filename,forwhat) VALUES
('motor.php3','Motor Tacho');
INSERT INTO files (filename,forwhat) VALUES
('op.php3','Operational Amplifier');
INSERT INTO files (filename,forwhat) VALUES
('out-pot.php3','Out Potentiometer');
INSERT INTO files (filename,forwhat) VALUES
('power.php3','Power Supply');
INSERT INTO files (filename,forwhat) VALUES
('pre.php3','Pre-Amplifier');
INSERT INTO files (filename,forwhat) VALUES
('servo.php3','Servo-Amplifier');
INSERT INTO files (filename,forwhat) VALUES
('signal.php3','Signal Generator or Source');
INSERT INTO files (filename,forwhat) VALUES
('cir1.php3','circuit connection for Experiment 1');
INSERT INTO files (filename,forwhat) VALUES
('cir2.php3','circuit connection for Experiment 2');
INSERT INTO files (filename,forwhat) VALUES
('cir3.php3','circuit connection for Experiment 3');
INSERT INTO files (filename,forwhat) VALUES
('cir4.php3','circuit connection for Experiment 4');
INSERT INTO files (filename,forwhat) VALUES
('cir5.php3','circuit connection for Experiment 5');

```

List A-2: MySQL codes for creating Table "files"

```

use elec344;
Drop table if exists visiting;

CREATE TABLE visiting (
ng tinyint(4) DEFAULT '0' NOT NULL AUTO_INCREMENT,
PRIMARY KEY (ng),
f1 tinyint(4) DEFAULT '0' NOT NULL,
f2 tinyint(4) DEFAULT '0' NOT NULL,
f3 tinyint(4) DEFAULT '0' NOT NULL,
f4 tinyint(4) DEFAULT '0' NOT NULL,
f5 tinyint(4) DEFAULT '0' NOT NULL,
f6 tinyint(4) DEFAULT '0' NOT NULL,
f7 tinyint(4) DEFAULT '0' NOT NULL,
f8 tinyint(4) DEFAULT '0' NOT NULL,
f9 tinyint(4) DEFAULT '0' NOT NULL,
f10 tinyint(4) DEFAULT '0' NOT NULL,
f11 tinyint(4) DEFAULT '0' NOT NULL,
f12 tinyint(4) DEFAULT '0' NOT NULL,
f13 tinyint(4) DEFAULT '0' NOT NULL,
f14 tinyint(4) DEFAULT '0' NOT NULL,
f15 tinyint(4) DEFAULT '0' NOT NULL
);
INSERT INTO visiting (f3) VALUES (0);
INSERT INTO visiting (f3) VALUES (0);

```


Appendix B : Table of Figures, Tables and Lists

B.1 Figures

Figure 2-1: A Simple Architecture of an Expert System.....	2-10
Figure 2-2: Typical Structure of an ITS.....	2-14
Figure 3-1: Laboratory Equipment	3-5
Figure 3-2: DACBoard Used for Digital Motor Control.....	3-6
Figure 3-3: “DigiLab” Package Used for Digital Motor Control	3-6
Figure 3-4: Overall Structure.....	3-7
Figure 3-5: Knowledge Domain Network	3-9
Figure 3-6: Various Scenarios of Access Designs for CALI-II.....	3-19
Figure 3-7: Flowchart for Circuit Simulation.....	3-20
Figure 4-1: Instruction Contents of CALI-II and Delivery Methods.....	4-2
Figure 4-2: A Pop-up Window with Three Frames	4-3
Figure 4-3: Multimedia Navigator II	4-7
Figure 4-4: RealEncoder.....	4-7
Figure 4-5: RealMedia Tools for Editing .rm Files	4-8
Figure 4-6: Flash Edit Environment with Timeline.....	4-9
Figure 4-7: Instance Properties Dialog Box & Library of Symbols	4-11
Figure 4-8: Movie Clip in Frame 5	4-11
Figure 4-9: Tell Target Action of a Button.....	4-12

Figure 4-10: GetURL Action of a Button.....	4-12
Figure 4-11: Aftershock Publisher.....	4-13
Figure 4-12: Scripts in the URL Area for the Get URL Action of the “Power Supply” Button	4-17
Figure 4-13: Scripts in the URL Area for the Get URL Action of the “Next” Button	4-17
Figure 4-14: Frame 2 in the Scene.....	4-21
Figure 4-15: Movie Clip “wire9”.....	4-21
Figure 4-16: Diagram Explaining How Movie Clip “wire9” Is Controlled	4-22
Figure 4-17: Scripts in the URL Area for the Get URL Action of the	4-23
Figure 5-1: User Authentication	5-2
Figure 5-2: Enrolment Form.....	5-3
Figure 5-3: Procedure in Checking the Enrolment	5-4
Figure 5-4: Procedure for Adding a User	5-5
Figure 5-5: Menu for Adding a User	5-5
Figure 5-6: Menu for Changing a User’s Details	5-6
Figure 5-7: Menu for Removing a User.....	5-6
Figure 5-8: Class Accesses or Login Records	5-7
Figure 5-9: Clicking a Group Number.....	5-8
Figure 5-10: Usage Log	5-8
Figure 5-11: Page Access Log on a Date.....	5-8

Figure 5-12: Contents in the “files” Table.....	5-9
Figure 5-13: Procedure Used in “Recorder1.inc”	5-14
Figure 5-14: Procedure Used in Student Guide.....	5-15
Figure 5-15: Feedback Given to Group 2 for the Access of “Cir3.php3”	5-16
Figure 5-16: Feedback Given to Group 3 for the Access of “Cir5.php3”	5-17
Figure 6-1: Login Page	6-2
Figure 6-2: User Interface for Student.....	6-2
Figure 6-3: Lecturer Navigation Frame	6-3
Figure 6-4: Student Navigation Frame	6-3
Figure 6-5: Equipment Menu.....	6-5
Figure 6-6: Component Pop-up Window	6-5
Figure 6-7: Pop-up Window for Tips	6-6
Figure 6-8: Pop-up Window for Circuit Diagram of Servo-Amp	6-7
Figure 6-9: Tags for Sockets.....	6-7
Figure 6-10: Pointing to Coloured Texts	6-8
Figure 6-11: Basic Connections.....	6-8
Figure 6-12: Project Procedure Menu.....	6-9
Figure 6-13: Pop-up Window for “What You Should Know”	6-10
Figure 6-14: Pop-up Menu.....	6-11
Figure 6-15: Procedure Pop-up Window with RealVideo Embedded.....	6-12
Figure 6-16: Pop-up Window for Component Choosing.....	6-12

Figure 6-17: Component Choosing Stages	6-13
Figure 6-18: Window for Circuit Connection.....	6-14
Figure 6-19: Tips for Circuit Connection	6-15
Figure 6-20: Pop-up Window for Troubleshooting	6-16
Figure 6-21: Troubleshooting Methods for “Motor doesn’t move”	6-16
Figure 6-22: Project Description	6-17
Figure 6-23: PDF Files from Course Notes Website.....	6-18
Figure 6-24: Pop-up Window for Resonant Peak.....	6-18
Figure 6-25: Forum.....	6-19
Figure 6-26: Column Chart of Student Survey Results for Q1~Q7	6-21
Figure 6-27: Pie Chart for Q1	6-22
Figure 6-28: Pie Chart for Q2.....	6-22
Figure 6-29: Pie Chart for Q3	6-23
Figure 6-30: Pie Chart for Q4.....	6-24
Figure 6-31: Pie Chart for Q5	6-25
Figure 6-32: Pie Chart for Q6.....	6-25
Figure 6-33: Pie Chart for Q7	6-26
Figure A-1: Directories Structure of CALI-II Which Should Be Set Up at the Base of the Server.	A-1
Figure A-2: Directory Holding the .cgi Files.....	A-1
Figure A-3: Directory Holding the .rm Files	A-1

B.2 Tables

Table 2-1: A Comparison of CAI and ITS	2-13
Table 4-1: Actions of Buttons “Terminal 1” and “Terminal 2” in Keyframes of Movie Clip “wire9”	4-22
Table 5-1: Contents in the Initial “visiting” Table	5-10
Table 5-2: Contents in the “visiting” Table.....	5-16

B.3 Lists

List 4-1: Comp2.js	4-15
List 4-2: /exp2/ex3.php3	4-18
List 4-3: Cir2.js.....	4-19
List 4-4: Equ.js.....	4-23
List 4-5: Equipment.php3	4-24
List 5-1: Recorder1.inc	5-12
List 6-1: Elec344 Student Survey	6-20

References

- Al-Jumeily, D.; Strickland, P. (1997). "Designing an interface on the Web for an intelligent tutoring System," *EUROMICRO 97. New Frontiers of Information Technology.*, Proceedings of the 23rd EUROMICRO Conference. Page(s): 158 – 162
- Alpert, D., & Bitzer, D.L. (1970). "Advances in Computer-Based Education," *Science*, 167, 1582-90.
- Allen Communication: Quest, (on-line). URL:
<http://www.allencomm.com/software/quest/>
- Allen, L.R. (1986). "Measuring attitude toward computer assisted instruction: the development of a semantic differential tool," *Computers in Nursing*, 4, 144-151.
- Alvey, P. (1982). "A programme for advanced information technology," HMSO, London
- Anderson J.R. (1982). "Acquisition of Cognitive Skills," *Psychological Review*, 89, 369-406
- Anderson J.R. (1985). "Skill Acquisition: compilation of weak method problem solutions," Carnegie-Mellon University
- Anderson J.R. (1988). The Expert Module. *Foundations of Intelligent Tutoring Systems*, Polson M.C. and Richardson J.J. (Eds), pp.21-53.
- Anderson J.R., Boyle C.F., and Yost G. (1985). "The Geometry Tutor," *Proceedings of IJCAI-85*.
- Anderson J.R. and Reiser B.J. (1985). "The LISP Tutor," *Byte*, Vol. 10, No. 4
- Anderson J.R., Boyle C.F. and Reiser B.J. (1985). "Intelligent Tutoring Systems," *Science* 228, 456-462

- Araki Y.; Imamura, H; Obo, T.; Kobori, Y. (1993). "Computer Assisted Instruction Learning System Using CAI," *Frontiers in Education Conference*, 1993. Twenty-Third Annual Conference. 'Engineering Education: Renewing America's Technology', Proceedings. Pages: 804.
- Arieatas R. (on-line). "Welcome to the OPTICS Laboratory Server," URL:
<http://www.uwi.tt/elec/lab/optics/>
- Bangert-Drowns, R. L.; Kulik, J. A.; and Kulik, C. C. (1985). "Effectiveness of Computer-Based Education in Secondary Schools," *Journal of Computer-Based Instruction*, 12/3, pp. 59-68.
- Barchan J., Woodmansee B.J. and Yazdani M. (1985). "A Prolog-base Tool for French Grammar Analysers," *Instructional Science*, Vol. 14
- Barr A. & Feigenbaum E.A. (1981). *The handbook of artificial intelligence*, Vol. 1, Los Altos, CA: William Kaufmann
- Batey, A. (December 1986). "Building a Case for Computers in Elementary Classrooms: A Summary of What the Researchers and the Practitioners Are Saying." Paper presented at *the Second Leadership in Computer Education Seminar*, Seattle, WA.
- Blunt L. (October 1998). "Web-based Asynchronous Virtual Classroom". Elec457 Thesis, School of Electrical, Computer and Telecommunications Engineering, University of Wollongong.
- Bolwell, C. (1988). "Evaluating Computer Assisted Instruction." *Nursing & Health Care*, 9, 511-515.
- Braun, L. (October 1990). "Vision: TEST (Technologically Enriched Schools of Tomorrow). Final Report: Recommendations for American Educational Decision Makers." *Eugene, OR: The International Society for Technology in Education*.

- Brown J.S. and Burton R.R. (1978). "Diagnostic models for procedural bugs in basic mathematical skill," *Cognitive Science*, 2, 155-192
- Brown J.S., Burton R.R. and de Kleer J. (1982). Pedagogical, natural language and knowledge engineering techniques in SOPHIE I, II and III in Sleeman and Brown (eds.).
- Burns H.L. and Capps C.G. (1988). Foundations of Intelligent Tutoring Systems: An Introduction. *Foundations of Intelligent Tutoring Systems*, Polson M.C. and Richardson J.J. (Eds).
- Burton R.R. (1982). DEBUGGY: Diagnosis of Errors in Basic Mathematical Skills in Sleeman and Brown (eds.).
- Burton R.R. and Brown J.S. (1982). An investigation of computer coaching for informal learning activities in Sleeman and Brown (eds.).
- Capper, J., and Copple, C. (1985). "Computer Use in Education: Research Review and Instructional Implications." Washington, DC: Center for Research into Practice.
- Carr B., and Goldstein I.P. (1977). "Overlays: A theory of modeling for computer aided instruction," Artificial Intelligence Laboratory Memo 406 (Logo Memo 40). Massachusetts Institute of Technology
- Chan W.L. and So A.T.P. (1994). "A Cost Effective Interactive Multimedia System for Electrical Undergraduate Laboratory Sessions", *Multi-Media Engineering Education Proceedings, IEEE First International Conference on, 1994*, Page(s): 219 -224
- Clancey W.J. (1979). "Tutoring rules for guiding a case method dialogue," *Int. J. of Man-Machine Studies*.

- Cuifo, P. (on-line 1). Welcome to the MiCE Board Homepage, URL:
<http://www.elec.uow.edu.au/mice/>
- Cuifo, P. (on-line 2). MiCE FTP Site, URL: <http://www.elec.uow.edu.au/mice/>
- Davies N. G., Dickens S. L. and Ford L. (1985). "TUTOR — A Prototype ICAI system" in Bramer M. (ed.), *Research and Development in Expert Systems* Cambridge University Press.
- Dean, C. and Whitlock, Q. (1983). "A Handbook of Computer-Based Training", Kogan Page, pp. 133-155.
- Dewey J. (1910). *How we think*. Boston: Heath.
- Edwards, J.; Norton, S.; Taylor, S.; Weiss, M.; and Dusseldorp, R. (1975). "How Effective is CAI? A Review of the Research." *Educational Leadership*, 33/2, pp. 147-153.
- Fletcher J.D. (1984). Intelligent instructional systems in training. In S. A. Andriole (Ed.), *Applications in artificial intelligence*. Princeton, NJ: Petrocelli
- Fletcher, J. D. (July 1990). "Effectiveness and cost of interactive videodisk instruction in defense training and education (p.2372). ". Washington, D.C.: Institute for Defense Analysis.
- French, D. (1986). "Using Learning Theory to Design and Evaluate Computer Assisted Instruction Software". *Nurse Educator*, 11, (5), 33-37.
- Gibson, E. J., Brewer, P. W., Dholakia, A., Vouk, M. A. & Bitzer, D.L. (on-line). A Comparative Analysis of "Web-Based Testing and Evaluation Systems." URL:
<http://renoir.csc.ncsu.edu/...rts/WebBased Testing.html>.
- Grimes, D. M. (1977) "Computers for Learning: The Uses of Computer Assisted Instruction (CAI). in California Public Schools". Sacramento, CA: California State Department of Education.

- Goldstein I.P. (1982). The genetic graph: A representation for the evolution of procedural knowledge. In D. Sleeman & J.S. Brown (Eds.), *Intelligent tutoring systems*. New York: Academic Press
- Hartley J.R. and Sleeman D.H. (1973). "Towards Intelligent Teaching Systems," *Int. J. of Man-Machine Studies*.
- Hasselbring, T. (1984). "Research on the Effectiveness of Computer-Based Instruction: A Review". *Technical Report* No. 84.1.3. Nashville, TN: George Peabody College for Teachers, Learning Technology Center, (ED 262 754).
- Hegarty C (1996). "Statutor: Intelligent Tutoring System?", BILETA'96 Conference Proceedings, 3 *The Journal of Information, Law and Technology (JILT)*. URL: <http://elj.warwick.ac.uk/elj/jilt/bileta/1996/3hegarty/>
- Howard, S. W., Ault, M. M., Knowlton, H. E., & Swall, R. A. (1992). "Distance education: Promises and caution for special education". *Teacher Education and Special Education*, 15, 275-283.
- Imlah W. and du Boulay B. (1986). *Robust Natural Language Parsing in Computer Assisted Language Instruction*, Cognitive Studies Programme, University of Sussex
- Johnson, M., K., & Amundsen, C. (1983). "Distance education: A unique blend of technology and pedagogy to train future special educators". *Journal of Special Education Technology*, 6, 34-45.
- Johnson W.L. and Soloway E. (1985). "PROUST," *Byte*, Vol., 10, No.4
- Kann, L. K. (1987). "Effects of Computer-assisted Instruction on Selected Interaction Skills Related to Responsible Sexuality". *Journal of School Health* 57/7, pp. 282-287.

- Kearsley G. (1987). *Artificial intelligence and instruction: applications and methods*. Addison-Wesley Publishing Company.
- Khoiny, F. E., (1995). "Factors that contribute to computer-assisted instruction effectiveness". *Computers in Nursing*, 13 (4), 165-168.
- Koffman E.B. & Blout S.E. (1975). "Artificial intelligence and automatic programming in CAI," *Artificial Intelligence*, 6, 215-234
- Kohl, J. E. & Su M.-C. (1995). "Computer Assisted Instruction: Implications for Achievement and Critical Thinking". (On-line). URL:
<http://www.nursing.ab.umd.edu/students/~jkohl/cai.htm>.
- Kolb, D. (1976). "Learning style inventory technical manual". Boston: Mcber and Company.
- Kozick R.J. and Crane C.C. (May 1996). "An integrated Environment for Modeling, Simulation, Digital Signal Processing and Control", *IEEE Trans on Education*, vol. 39, no. 2.
- Kulik J.A, Bangert R.L. & Williams G.W. (1983). "Effects of Computer-Based Teaching on Secondary School Students" in *Journal of Educational Psychology*, 75, 19-26.
- Kulik, J. (April 1985). "Consistencies in Findings on Computer-Based Education". Paper presented at the *Annual Meeting of the American Educational Research Association*, (ED 263 880).
- Kulik, J. A. (1983). "Synthesis of Research on Computer-Based Instruction." *Educational Leadership* 41/1, pp. 19-21.
- Kulik, J. A.; Bangert, R. L.; and Williams, G. W. (1983). "Effects of Computer-Based Teaching on Secondary School Students". *Journal of Educational Psychology* 75/1: 19-26.

- Kulik, J. A., and Kulik, C. C. (February-March, 1987). "Computer-Based Instruction: What 200 Evaluations Say". Paper presented at *the Annual Convention of the Association for Educational Communications and Technology*, Atlanta, GA, (ED 285 521).
- Kulik, J. A.; Kulik, C. C.; and Bangert-Drowns, R. L. (1985). "Effectiveness of Computer-Based Education in Elementary Schools." *Computers in Human Behavior* 1/1: 59-74.
- Lawler R. (1984). Designing Computer-based Microworlds in Yazdani (ed.).
- Macromedia, Flash Help Pages, available from the Flash 3 package.
- Macromedia Director, (on-line). URL: <http://macromedia.com/software/director/>
- Macromedia Flash, (on-line 1). URL: <http://www.macromedia.com/software/flash/>
- Macromedia Flash — Support Overview. (on-line 2). URL: <http://www.macromedia.com/support/flash/>
- Marshall, D. (1988) CAL / CBT - The great debate, Computer-Based Training (CBT), Chartwell-Bratt, pp. 218-234.
- Martin, G. R. (1973). TIES Research Project Report: The 1972-73 Drill and Practice Study. St. Paul, MN: Minnesota School District Data Processing Joint Board.
- McCalla G.I. and Greer J.E. (1991). "Granularity-Based Reasoning and Belief Revision in Student Models," *The Key to Individualised Knowledge-Based Instruction*, Greer J.E. and McCalla G. I. (Eds), Nato ASI Series.
- Miller M.J. (March 31, 1992). "Multimedia," *PC Magazine*, Vol. 11, No. 6, pp. 112-123.
- Minsky M. (1975). A framework for representing knowledge. In P. H. Winston (Ed.), *The psychology of computer vision*. New York: McGraw-Hill.
- MySQL, (on-line). URL: <https://www.mysql.com/>

- Nagel L.W. and Pedersen D.O. (1973). "Simulation program with integrated circuit emphasis," *Proceedings of the 6th Midwest Symposium Circuit Theory*, Waterloo, Canada
- Naghdy F. and Dedhia D. (February 1997). "Computer Aided Instruction in Control Laboratory", *Proceeding of the Pacific Region Conference on Electrical Engineering Education. PRCEEE'97*, Wollongong, Australia.
- Naghdy F. (on-line). Course Materials, ELEC344 - Classic Control, Control Systems, URL: <http://www.elec.uow.edu.au/courses/elec344/classic.htm>
- Neil, R. (1985). "Effects of computer-assisted instruction on nursing student learning and attitude". *Journal of Nursing Education*, 24 (2), 72-75.
- Newell A. and Simon H.A. (1972). Human problem solving. Englewood Cliffs, NJ: Prentice-Hall
- Nichols, G. (1995). (On-line). "Formative Evaluation of Web Based Training". URL: <http://ucalgary.ca/~gwnichol/formeval/formeval.html>.
- Norman D.A. & Rumelhart D.E. (1975). "Explorations in Cognition," San Francisco: Freeman
- O'Shea T. and Self J. (1983). "Learning and Teaching with Computers," Harvester Press
- O'Shea T., Bornat, R., du Boulay, B., Eisenstad M. and Page. I. (1984). Tools for Creating Intelligent Computer Tutors in Elithor and Banerjii (eds). Human and Artificial Intelligence North Holland
- Omniboard, (on-line). URL: <http://www.omnitalk.com/download/omniboard.html>
- Papert S. (1980). "Mindstorms, Children, Computers and Powerful Ideas," Harvester Press/Basic Books

- Parks, E. (1996). "Computer Based Training (CBT). Evaluation Tool". Lakewood's Technology for Learning. 4.
- PHP: Hypertext Preprocessor, (on-line). URL: <http://www.php.net/>
- PHPBuilder.com, (on-line). The Resource for PHP Developers, URL: <http://www.phpbuilder.com/>
- Powell A. (on-line). "Adam's Multimedia Tutorial". URL: <http://hotwired.lycos.com/webmonkey/multimedia/tutorials/tutorial3.html>
- Pressey S.L. (1927). "A machine for automatic teaching of drill material," *School and Society*, 25, 1-14
- Quillian M.R. (1968). Semantic memory. In M. Minsky (Ed.), *Semantic information processing*. Cambridge, MA: MIT Press
- Rapaport, P., and Savard, W. G. (1980). "Computer-Assisted Instruction," Topic Summary Report. Portland, OR: Northwest Regional Educational Laboratory, (ED 214 707).
- Real.com, (on-line). URL: <http://www.real.com/>
- Rickel, J.W. (1989). "Intelligent Computer-Aided Instruction: A survey organized around system components," *IEEE Transactions on Systems, Man and Cybernetics*, Volume 19, Number 1, January/February 1989, pp. 40-57.
- Rodgers J., (1999). "Web-Based Instruction and Assessment of Control Laboratory," Elec457 Thesis, School of Electrical, Computer and Telecommunications Engineering, University of Wollongong.
- Rupe, V. S. (1986). "A Study of Computer-Assisted Instruction: Its Uses, Effects, Advantages, and Limitations". South Bend, IN: Indiana University, (ED 282 513).
- Saba, V.K. & McCormick, K.A. (1996). "Essentials of computers for nurses," (2nd ed.). New York: McGraw Hill.

- Sabitzer, T. (October, 1998). "Digital Control Laboratory Based on Mice Board", Undergraduate Thesis, School of Electrical, Computer and Telecommunications Engineering, University of Wollongong.
- Salomon, G., Perkins, D. N., & Globerson, T. (1991). "Partners in cognition: Extending human intelligence with intelligent technologies". *Educational Researcher*, 20 (4), 2-9.
- Schank R.C. and Abelson R.P. (1977). "Scripts, plans, goals, and understanding: An inquiry into human knowledge structures," Hillsdale, NJ:Lawrence Erlbaum Associates
- Schmidt, .S. M., Arndt, M. J., Gaston, S., & Miller, B. J. (1991). "The effectiveness of computer-managed instruction versus traditional classroom lecture on achievement outcomes," *Computers in Nursing*, 9 (4), 159-163.
- Shlechter, T.M. (1991). "Problems and promises of Computer-Based Training", Promises, promises, promises: History and foundations of Computer-Based Training, Ablex Publishing, pp. 1-17.
- Shortcliffe E.H. (1976). *Computer-based Medical Consultations: MYCIN* American Elsevier
- Skinner B.F. (1958). "Teaching Machines," *Science*, Vol. 128
- Sleeman D. (1982). Assessing aspects of competence in basic algebra. In D. Sleeman & J> S. Brown (Eds.), *Intelligent tutoring systems*. New York: Academic Press
- Sleeman D. and Brown J.S. (eds.). (1982). "Intelligent Tutoring Systems," Academic Press
- Specht, M. (1998). "ATS-adaptive teaching system a WWW-based ITS", *Web-Based Knowledge Servers (Digest No. 1998/307)*, *IEE Colloquium on, 1998*, Page(s): 6/1 -6/6.

- Springer, S., & Deutsch, G. (1981). "Left brain, right brain," San Francisco: W. H. Freeman and Company.
- Stauffer, K. (1996). "Student Modeling and Web-based Learning Systems". (On-line). URL:
<http://ccism.pc.athabascau.ca.html/...age/project/initsm.htm#introduction>
- Steinwand D. (On-line). "Welcome to the CALI Project",
<http://www.intranet.csupomona.edu/~cali/>
- Stelzer J. & Garneau J. (1972). "Project IMPACT software documentation: Overview of the computer-administered instruction subsystem," Technical Report 72-21. Alexandria, VA: Human Resources Research Organization.
- Tira A.S. (October 1998). "Computer Assisted Laboratory Instruction". Undergraduate Thesis, School of Electrical, Computer and Telecommunications Engineering, University of Wonllongong.
- Treuhft, J. (1995). "Changes in education". (On-line). URL:
<http://www.algonquinc.on.ca/edtech/change.html>.
- University of Maine. (1995). Courseware Evaluation. (On-line). URL:
<http://kramer.ume.maine.edu/cev/1.html>.
- VanLehn K. (1988). Student Modeling. *Foundations of Intelligent Tutoring Systems*, Polson M.C. and Richardson J.J. (Eds).
- Vitale, B. (1981). "Unicorns are real: a right-brained approach to learning". Rolling Hills, CA: Jalmar Press.
- Uttal W.R., Rogers M., Hieronymous R., & Pasich T. (1969). "Generative computer-assisted instruction in analytic geometry," Newburyport, MA: Entelek, Inc.

- Warendorf, K. (1997). "ADIS - An Animated Data Structure Intelligent Tutoring System on the WWW", *International Conference on Information, Communications and Signal Processing, ICICS'97*, Singapore, 9-12 September, 1997.
- Michael K. (on-line). Intro to Flash 3, Webmonkey, URL:
<http://hotwired.lycos.com/webmonkey/98/25/index1a.html?tw=multimedia>
- Whittington, D. (1996). "Some Issues raised by a Virtual University". (On-line).
URL: <http://www.epcc.ed.uk/WWW6workshop/submission/position.html#third>.
- Wingrad T. (1975). Frame representation and the declarative-procedural controversy.
In D. G. Borow & A. Collins (Eds.), *Representation and understanding: Studies in cognitive science*. New York: Academic Press.
- Wong, T. K. (1990). "Drug calculations for nurses-A computer assisted learning application". *Nursing Education Today*, 10, 274-280.
- Wright M. (On-line). WWWBoard, URL:
<http://worldwidemart.com/scripts/wwwboard.shtml>
- Yazdani M. (On-line). "Intelligent Tutoring Systems: An overview." Department of Computer Science University of Exeter. URL:
<http://www.intellectbooks.com/authors/lawler/its.htm>
- Yazdani M. (ed.). (1984). "New Horizons in Educational Computing," Ellis Horwood/John Wiley.