

ADDIS ABABA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

SCHOOL OF INFORMATION STUDIES FOR AFRICA

A HYPERTEXT BASED COMPUTER AIDED INSTRUCTION SYSTEM

IN AMAHARIC:

A CASE STUDY WITH THE COURSE 'INTRODUCTION TO COMPUTERS'

**A THESIS SUBMITTED IN PARTIAL FULFILMENT OF
THE REQUIREMENT FOR THE DEGREE OF MASTER OF
SCIENCE IN INFORMATION SCIENCE**

BY

GIRMA DESSALEGN

May 1996

ADDIS ABABA UNIVERS
LIBRARIES
P.O. BOX 1176
ADDIS ABABA ETHIOPIA

ADDIS ABABA UNIVERSITY
SCHOOL OF GRADUATE OF STUDIES
SCHOOL OF INFORMATION STUDIES FOR AFRICA


A HYPERTEXT BASED COMPUTER AIDED INSTRUCTION SYSTEM
IN AMHARIC:
A CASE STUDY WITH THE COURSE INTRODUCTION TO COMPUTERS.

By

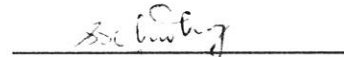
Girma Dessalegn

Name and Signature of Members of the Examining Board

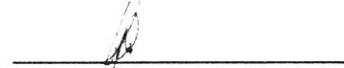
Ato Getachew Birru, Chairman, Examining Board



Dr. G.G Chowdhury, Advisor



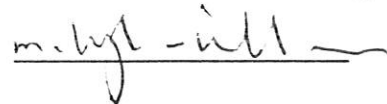
Ato Nega Alemayehu, Advisor



Ato Tesfaye Biru, Internal Examiner



Mr. Michael Lloyd-Williams, External Examiner



DEDICATION

TO MY MOTHER ZEWDIE YIMER

FATHER DESSALEGN MEKONNEN

SISTERS ASELEFECH, SERKALEM, YELIBE & BEGIZESH

BROTHER MULUGETA

ACKNOWLEDGEMENT

I would like to thank Dr. G. G. Chowdhury and Ato Nega Alemayehu who have been very encouraging and stimulating supervisors; Ato Mulugeta Libssie, Ato Negusu Atnafu, and Ato Getachew Mengistu for assisting and providing the English and Amaharic versions of the Introduction to Computers course material used in this study; Ato Gobena Mikael for his invaluable support in many ways, some of which were very decisive for the actualization of this thesis.

I am also indebted to Ethiopian Science and Technology Commission for letting me study; Ato Mulu G/Egziabher and NCIC library staffs for their help in searching for documents; Ato Sultan Mohammed and Ato Dejene Ejigu for their encouragements and constructive comments; My sisters W/t Yelibe and W/t Abenet for their unfailing care and patience throughout my study; the Academic and support staffs of SISA and to all my friends.

ABSTRACT

The application of computers to the process of teaching and learning holds many promises. Their use has been seen as a way of improving the quality of delivery, increase the individualization of learning, encouraging learning privacy, provide greater motivation, improve information retention. Other advantages include immediate feedback, the opportunity of lessable students to benefit from repeated use, and more active participation by students. Of all the possible applications of computer technology to learning, hypertext seems to be one with a great potential. Hypertext allows learners to access randomly stored information based on their needs, interest, or whims. It focuses learner attention on the relationship of facts. It also encourages active student-centred learning since there are always decisions about what to read next. Learning in one's own language is believed to be important. One of the most important advantages of learning in one's own language is that learning begins with something that is already known. The need for training on the basics of computers is strongly felt in Ethiopia. This high demand of producing computer literate people is being responded by the mushrooming of computer training institutions in Addis Ababa and the major cities of the country. It was identified that the training situation, being in its infancy, is faced with many problems.

This study attempts to demonstrate the design and development of a hypertext based courseware in Amaharic by considering a case study on the course 'Introduction to Computers' which is being offered at the National Computer and Information Centre of Ethiopian Science and Technology Commission. It also tries to address problems regarding the computer training situation in the country.



A literature review on the definition, advantage, design consideration, types, and current trends and developments on hypertext and courseware is presented. Different models of hypertext and courseware development are discussed. The models are used to identify the steps and processes involved in the development process of a hypertext courseware which satisfies instructional system principles. The quick prototyping software system development approach is used to develop the courseware. KnowledgePro for Windows is used to develop the program.

The result of the study indicates:

1. There is a need for training on the basics of computers in the country. The training which is being provided is not that much satisfactory mainly because of a shortage of properly trained instructors among other reasons. This necessitates the introduction of courseware. It was also indicated that developing courseware in Amaharic is advantageous.
2. It is technically possible to develop Amaharic applications using the quick prototyping approach and programming environments capable of displaying Amaharic fonts at runtime, even though the currently available program editors cannot display Amaharic characters.

TABLE OF CONTENTS

DECLARATION.....	i
DEDICATION.....	ii
ACKNOWLEDGEMENT	iii
ABSTRACT	iv
TABLE OF CONTENTS.....	vi
LIST OF TABLES	x
LIST OF FIGURES.....	xi

CHAPTER ONE

INTRODUCTION.....	1
1.1 Background.....	1
1.2 Statement of the Problem	3
1.3 Justification.....	5
1.4 Objective	10
1.5 Scope and Limitations	11
1.6 Methodology.....	12
1.6.1 Data Collection	12
1.6.2 Development Method	13
1.7 Organization of Thesis.....	13

CHAPTER TWO

THE COURSE: 'INTRODUCTION TO COMPUTERS'.....	15
2.1 Introduction.....	15

2.2	Objectives	17
2.3	Entrance Requirement.....	17
2.4	Duration.....	18
2.5	Course Content.....	18
2.6	Instructors.....	20
2.7	Problems.....	21

CHAPTER THREE

COURSEWARE.....	22
-----------------	----

3.1	Computers in Education.....	22
3.2	Definition.....	23
3.3	Advantages of Computer Aided Instruction.....	25
3.3.1	Increased Control.....	26
3.3.2	Reduced Resources.....	27
3.3.3	Individualization.....	28
3.3.4	Improved Student Performance.....	29
3.3.5	Increased Learner Satisfaction.....	30
3.3.6	Convenience.....	30
3.3.7	Power of the Computer.....	31
3.4	Types of Courseware.....	32
3.5	Courseware Design Characteristics.....	36
3.6	Current Trends and Developments.....	38
3.6.1	Hypermedia.....	39
3.6.2	Expert Systems.....	41
3.6.3	Expertext.....	44
3.6.4	Object Oriented Programming.....	45

CHAPTER FOUR

HYPertext	48
4.1 Introduction.....	48
4.1.1 Linear Structures.....	48
4.1.2 Non-linear Structures.....	49
4.2 Definition.....	51
4.3 Historical Background.....	53
4.4 Types of Hypertext.....	58
4.5 Authoring Tools.....	59
4.5.1 KnowledgePro.....	60
4.6 Educational Application.....	61
4.7 Current Trends and Developments.....	64
4.7.1 Pointing Devices.....	64
4.7.2 Hypertext and Artificial Intelligence.....	65
4.7.3 Hypertext and Telecommunication.....	66

CHAPTER FIVE

DEVELOPMENT OF A HYPertext BASED COURSEWARE.....	68
5.1 Design Considerations.....	68
5.2 Node Design.....	70
5.2.1 Language Usage.....	71
5.2.2 Highlighting Techniques.....	72
5.2.3 Operating Directions.....	73
5.2.4 Screen Layouts.....	74
5.3 Link Design.....	75
5.3.1 Content Organization.....	76
5.3.2 Presentation Flow.....	79
5.4 Development Models.....	82
5.4.1 Using Prototypes.....	90

CHAPTER SIX

PROTOTYPE DEVELOPMENT AND IMPLEMENTATION 94

6.1	The Definition Phase: Requirement Analysis.....	94
6.2	The Development Phase.....	97
6.2.1	Quick Design.....	97
6.2.2	Prototype Development.....	102
6.3	Implementation.....	116

CHAPTER SEVEN

CONCLUSION AND RECOMMENDATIONS 118

7.1	Conclusion.....	118
7.2	Recommendations.....	121
7.2.1	For Maturing the courseware.....	121
7.2.2	For further study.....	123

BIBLIOGRAPHY 125

APPENDIX 133

LIST OF TABLES

4.1	Hypertext Timeline	54
5.1	Internal Learning Needs with External Instructional Events	80

LIST OF FIGURES

3.1	Representation of the basic concept of education.....	23
5.1	Three basic organizations of hypertext courseware.....	78
5.2	Presentation flow with instructional events.....	82
5.3	Courseware Design Model.....	82
5.4	Phase 1: Preparation and Planning.....	83
5.5	Phase 2: Predesign.....	83
5.6	Phase 3: Design.....	83
5.7	Phase 4: Programming and formative evaluation.....	84
5.8	Phase 5: Summative Evaluation by Means of CMI.....	84
5.9	The Team approach to CBL courseware design.....	85
5.10	Overall Architecture of the CACE.....	86
5.11	Major phases of ISD.....	87
5.12	Analysis Phase.....	87
5.13	Design Phase.....	87
5.14	The Development Phase.....	88
5.15	The Implementation Phase.....	88
5.16	Basic hypertext development model.....	89
5.17	Prototyping.....	92
6.1	The structure of the courseware.....	102
6.2	Development flowchart of the courseware.....	103
6.3	Linking between screens for the first lesson.....	104
6.4	Title Screen.....	111
6.5	Objectives Screen.....	111
6.6	Contents Screen.....	112
6.7	Main Topic Objectives Screen.....	112
6.8	Module Overview Screen.....	113
6.9	Module elaboration with links to related items Screen.....	113
6.10	A last screen on a topic with a link to exercises.....	114

6.11	Exercise Screen	114
6.12	Information Screen to repeat a topic	115
6.13	Information Screen to proceed to next lesson	115
6.14	Information on incorrect answers with links to appropriate topics	115

CHAPTER ONE

INTRODUCTION

1.1 Background

Computer technology has influenced every aspect of modern life. There are thousands of areas currently influenced by such technology and new applications are being developed from time to time. Among the areas that use computer technology a great deal are business and industry, government, and medicine. Scientists, educators, and artists also employ computer to assist them in their work and to enable them use their time more creatively. In addition, computer technology has had a significant effect on entertainment and arts. Computers have even begun to influence the home lives of their users (Lavrine and Fecher 1986).

Computer applications in the field of education have been the following:

- the general use of computers and high level programming languages in problem solving;
- computerized tutorials;
- computerized drill-and-practice sessions; and
- computerized simulations and games,

with the last three being most commonly used in education and training.

Computer Aided Instruction (CAI), use of computers in educational process, can be used:

- to help students accomplish certain tasks, such as wordprocessing, numerical analysis, etc.;
- writing programs to direct the computer; and
- to present the materials to be learned by the students. The program which delivers the education is called a courseware (Hendrickson, Janson, and Rossi 1985, Alemayehu 1995).

The first implementations of CAI were simply "books on screen". The first and most common form was linear, where the student was presented with a sequence of displays consisting of text and graphics which defined the lesson and group of questions and feedback responses that would evaluate the student's understanding of the topic (Hendrickson, Janson, and Rossi 1985). With time, the learning process was enhanced by presenting the lesson in a non-linear form.

When students are trained with a computer-based training in a non-linear condition, i.e. in an environment where students have the freedom to navigate through different instructional and practice modules, they significantly improve their performance than in a linear condition, i.e. a training in a fixed format. It is suggested that this freedom of navigation and flexibility can be achieved by hypertext (Staton and Stammers 1990).

Hypertext can simply be defined as a system to manage a collection of information that can be accessed non-sequentially. It consists of chunks or fragments of text or other information. Nodes and associative links are the basic building blocks of all hypertext

systems. Rather than a continuous flow of text, hypertext breaks it up into units or modules of information. Because of the modularization of text, nodes may be accessed in any sequence that meets the information needs of the user. Links connect the various chunks or nodes and enable the user to determine the order in which information is presented. They enable the user to move around the hypertext. Links also provide connections between documents which can be in different media-written text, graphic text of various kinds, and sound. Hypertext tends to be under-directed, allowing the user to pick and choose his or her own path through the information map, ignoring some things and exploring others in depth (Lucarell and Zanzi 1993).

Hypertext has shown a great deal of promise in a number of areas such as text authoring, on-line information retrieval, on-line help for software systems, computer-based instruction, and even computer aided software engineering (CASE) (Parsaye et al. 1989).

1.2 Statement of the Problem

In Ethiopia computers are being used in different government, international, and business organizations. However, studies have shown that computers were underutilized in Ethiopia (Sirak 1988, Research & Planning Directorate 1990, and Rahel 1992). According to Research & Planning Directorate (1990) Some of the major reasons for the underutilization of computers may be attributed to:

- inadequate feasibility study;
- inadequate quality of computer staff; and

- lack of appreciation of the services by the top management.

Provision of adequate training to computer staff and management could solve the problem partially. However, provision of adequate training calls for a number of resources, the most important being properly trained educators/instructors. This is one of the major problems at the moment in Ethiopia.

Appropriate measures for developing courseware in different areas of IT training could alleviate the problem to a large extent. In this research an attempt will be made to introduce the benefits of using a courseware in a computer training programme through the medium of one of the written languages of the country.

Although it is possible to acquire courseware in one of the two ways (i.e. through purchase or inhouse development), the inhouse development option is chosen in view of the following considerations:

- the software's specialized nature;
- degree of customization required;
- lack of awareness of the potentials of courseware;
- availability of skill; etc.

The goal of this research is, therefore, to show the possibility of developing a hypertext-based courseware in Amaharic with a case study of the course '**Introduction to Computers**' which is being offered at the National Computer & Information Center of Ethiopian Science & Technology Commission.

1.3 Justification

Currently, business and government organizations are aware of the need for training and retraining of their staff in the field of computers. Some are even specifying computer skill as a prerequisite for employment. As a result, computer training institutions are mushrooming in Addis Ababa and in some of the major cities.

On the other hand, surveys have indicated that except those students who are actually engaged in the profession, the rest of the students are left with confused minds and their certificates (Sirak 1988, Research and Planning Directorate 1990). According to Azeb (1986) and Asmerom et al. (1989), the problems can be categorized into two: problems related to the institutions teaching methodology and problems related to pre-college factors.

Some of the problems could be related to:

- the emphasis of training centers being on the instructors' knowledge of the subject matter rather than the way they present the subject matter for trainees;
- students' difficulty to understand new concepts in a language other than their mother tongue; and

- composition of students with a varying degree of level of education in the same class.

There are some suggested solutions to improve the teaching process (Azeb 1986, and Asmerom et al. 1989). The situation could become effective if:

- student-centered teaching is followed;
- the nature of difference among students is recognized;
- students' needs are met by diagnosing learning difficulties;
- student interaction is active;
- flexibility is in place;
- questioning is introduced; and
- pace of learning of the student is considered.

Education specialists were predicting extensive reliance on CAI since the 1980s for effective learning. As Bork (1981) reports: "Eric Ashby, the British technologist, speaks of three great revolutions in human education: the forming of schools, the use of the written medium, and the invention of books. He then goes on to talk about fourth revolution. This revolution is the improvement in learning to be affected by modern technology - the electronic media and particularly computers."

One of the most widely accepted advantages of CAI is that it makes learning an active process. Most conventional educational methods (i.e. classes, lectures, and readings) allow

the students to adopt a passive attitude. Most psychologists agree that active involvement facilitates learning (Hendrickson, Janson, Rossi 1985).

It is also important in encouraging students to be self-directed and independent, allowing students to work at their own pace and in their own time, as well as to review their work. CAI provides great advantages for both slow learners and gifted ones. The development of students' confidence and skill in using computers and in learning how to cope with new technologies can also be achieved (Kirk 1993).

Another important advantage as Main (1984) [quoted in Hendrickson, Janson, and Rossi (1985)] put it "the systems are patient, never have a day off, don't loose tempers, and have all the time in the world".

As Alemayehu (1995) stated, situations where courseware are to be of assistance are:

- Extensive individual student practice is needed;
- Students' future job involves the daily use of computers;
- Safety is a matter of concern;
- Students' motivation is a matter of concern; and
- The material is very hard to teach by other methods.

The current status of computer training institutions in Ethiopia is the ideal situation to which a courseware could be beneficial.

In this research hypertext will be used as a tool to develop the courseware.

Hypertext is a tool for building and using associative structures. A normal document is linear, and one tends to read it from beginning to end. In contrast, reading a hypertext document is open-ended and one can jump from idea to idea depending on one's interests (Parsaye et al. 1989).

Computers do a good job of storing, retrieving and presenting vast quantities of data. The human mind is far more facile at filtering data to access information appropriate for the job at hand than are computers.

Hypertext builds upon the relative strengths of the human mind and the digital computer: the computer holds the data and presents to the human, the human chooses which way to go by pointing at each juncture. It is an intuitive approach in which the ability to link associated text matches the brain's natural tendency to think associatively (Berk and Devlin 1991).

Since Amaharic is the Federal working language of Ethiopia; medium of instruction in most schools; medium of communication in business; and has its own scripts, it could have a positive impact on the learning process, had courses been given in Amaharic.

As Poth (1980) stated, languages that stem from the environment are rich and efficient enough for conveying information drawn from the environment. They are the best tool for transmitting basic knowledge and skills.

The view that education is best given in the mother tongue seems to be widely accepted among educators. One of the most important advantages of learner's mother tongue is that learning begins with something that is already known by the learner. It has all resources necessary for the expression of any shade of meanings (Morsey and Limage 1987).

UNESCO has also recommended that pupil should begin their schooling through the medium of native tongue and its use should be extended to as late a stage in education as possible.

IDRC (International Development Research Center, Canada) has also to say something about this: "The promotion of mother tongue in school education, by and large, creates interest in learning, contributes to lower dropout-rate, and instills pride in traditional culture and oral literature and act as a motivator. This inclusion of language in school curriculum raises the dignity and adds to the prestige of those who speak it" (IDRC 1979).

In addition to making the teaching situation effective, the development of the courseware in Amaharic will increase the application of the Amaharic language in the computer field.

So far some developments have taken place in the application of the Amaharic language in the computer field. These developments include: a complete Amaharic interface based on an Amaharic Operating System-Agafari; add-in programs on major application packages, such as Wordperfect 5.1, most Windows based application programs, etc.

The course '**Introduction to Computers**', which is being offered in the National Computer and Information Center (NCIC) of Ethiopian Science and Technology

commission since 1988, and in all the other training institutions since their establishment, is selected as a case problem mainly because of the following reasons:

1. As an instructor of the course at NCIC, the researcher has adequate knowledge of the subject matter, and can easily get additional information and existing problems about the subject from senior staff if needed.
2. The researcher's experience in teaching the course for seven years has indicated that usually students come with the expectation of some interaction with the computer in learning a computer course. They are mostly bored with the subject since it is based entirely on lectures and manuals.
3. The students are admitted to the course with their varying educational background. Therefore, it is not easy to teach students in a uniform pace.
4. The course content do not change frequently as in the case of application software packages which change with new releases.
5. The course is basic for introducing users to the field of computers and if they are not motivated in the first course they are likely to loose interest for the remaining courses.

Furthermore, the experience gained in developing this courseware could be used to develop other courseware, help facilities and tutorials for software programs developed at NCIC where the researcher is a staff member.

1.4 Objectives

The general objective of this study is to develop a prototype hypertext-based courseware to teach the basics of computers in Amaharic.

The specific objectives of the project are to:

- identify common problems of trainees;
- provide an overview of hypertext systems and their applications with special reference to computer-based instruction;
- describe courseware types and their respective characteristics;
- provide a highlight on current trends and developments in courseware design;
- describe some of the available courseware development methodologies;
- demonstrate the development of a hypertext based courseware using the course **'Introduction to computers'** in Amaharic ; and
- demonstrate some of the features of the designed courseware.

1.5 Scope and Limitations

The scope of the study is limited to a brief description on some of the available courseware and hypertext development methodologies. In addition, development of a mature and fully developed Amaharic courseware is beyond the scope of this work. This is mainly due to

time constraint which makes it impossible to expose the courseware to the target students and get the feedback for possible modification.

Courseware development requires a team effort from a variety of disciplines. In this regard the researcher has consulted people involved in different fields. Software experts at the National Computer and Information Center, the thesis advisors and some SISA graduates are among the people consulted. Some hypertext-based software help programs and some tutorials are studied to broaden the researcher's knowledge in these areas. However, the courseware may still have some limitations due to lack of specialist resource persons in some of the fields.

1.6 Methodology

1.6.1 Data Collection

To obtain information pertinent to the formulation of the research problem, data has been gathered from formal and informal discussions with other instructors, interview with persons from training institutions, results of students' course assessment at the end of courses (this was not done for this research), and the researcher's experience.

After identification of the problems related to the training program on '**Introduction to Computers**' and deciding to introduce a Computer-Aided-Instruction (CAI) data has been collected from literature published on courseware development.

Of all the applications of computer technology in education, hypermedia seems to be one with a great potential (Bell 1993). To explore the possibilities of making the courseware hypertext-based, literature related to hypertext have been reviewed.

1.6.2 Development Method

The production of a successful hypertext-based courseware depends critically upon the adoption of a sound and systematic design strategy that is based upon the use of well-founded guidelines and established design procedures.

Alemayehu (1995) discusses some courseware development models, Barker (1993) examines some hypermedia design issues, Yang and Moore (1995) provides presentation flow for hypertext-based courseware. In this research after giving a brief overview of some of the models on courseware and hypertext development, a combination of the processes used for hypertext and courseware is used. The prototyping system development approach is used to test the screen layouts, colour combinations, button designs, link strategies, interaction methodologies and various other design parameters.

1.7 Organization of Thesis

The thesis is divided into seven chapters. Chapter one gives the introduction. Chapter two discusses the course '**Introduction to Computers**' which is the subject matter of the courseware. In chapter three courseware is discussed, its definition, the different types of courseware, some design characteristics, and current trends and developments are covered.

Definition of hypertext, types of hypertext, its application in education, tools for authoring hypertext, and current trends and developments in hypertext are discussed in chapter four. Chapter five covers the development considerations of a hypertext-based courseware. Chapter six deals with the development of the prototype courseware and the implementation plan. The seventh chapter presents concluding remarks and recommendations. Finally comes the bibliographic references, and sample tests and lessons of the prototype system are annexed.

CHAPTER TWO

THE COURSE: 'INTRODUCTION TO COMPUTERS'

2.1 Introduction

The National Computer and Information Center (NCIC) is a government institution resulted from the merging of the former National Computer Center and the National Scientific and Technological Information and Documentation Center which were responsible, since mid 1987, for the promotion of computer technology and information systems and services in the country, respectively.

Some of the objectives of the Center include building up national capabilities in the efficient utilization and development of computer and information science and technology in the areas of information systems and services, computer hardware and software, training, consultancy, and data communication (NCIC 1995).

The Center provides services, such as, computer services, computerized information search services, consultancy services, e-mail and BBS services, library services, training services, etc.

The National Computer Center (NCC), now NCIC, has started providing short term training courses since 1988. The computer training unit has the objective of training

trainees in hardware, software (operating systems, application software packages, programming languages), systems analysis and design methodologies, management of computer-based information systems and services, data communication and networking.

Introduction to computers is one of the prerequisites of the fulfillment of the Certificate programs in any one of the applications at NCIC.

Treainees upon securing one of the Certificates, i.e., after taking '**Introduction to Computers**', MS-DOS/Windows, and one of the application Programs in Wordprocessing (WordPerfect or Word), Spreadsheet (Lotus 1-2-3 or Excel), and database management (dBASE or FOX Pro) are expected to achieve the following basic goals of the unit.

The basic goals of the unit states that: a trainee, after completing the three months course in one of the applications, is expected to be able to handle most of the problems in the areas of word processing, database management, spreadsheet (or all) of an organization efficiently.

Even though the exact figure is not known, the Center has trained more than 2500 trainees so far.

2.2 Objectives

The course has the following objectives:

- to discuss the basic elements of data processing activity and why it is necessary;
- to introduce trainees to concepts and terminologies in the field of computers;
- to make trainees understand what computers can do and what they cannot do;
- to teach trainees the different constituent parts of a computer system (hardware, software); and
- to introduce trainees to the concepts of networking.

2.3 Entrance Requirement

Trainees who are registered for this course are not expected to have any prior knowledge of computer systems. The minimum requirement is to complete their high school education and proficiency of the English language.

2.4 Duration

The course is given for 18 hours. It is given in every other day basis in two groups.

Three classes are given in a day. Two of them in the day session (morning and afternoon) and one in the evening session. The day sessions are given 3 hours per day and the evening sessions 2 hours per day. Therefore, the course takes 3 to 4 weeks including examination periods.

2.5 Course Content

The course is divided into five sections as follows. The first section discusses about data processing. Section two deals with computers and how data is stored, processed and communicated in computers. Section three addresses the hardware aspect of the computer system and section four addresses the software aspect. Computer Communications is introduced in section five. Topics covered under each section are mentioned below.

1. Introduction
 - What this course is all about
 - What is data processing
 - Data processing criteria
 - Different ways of data processing

2. About computers

- What is a Computer
- The computer number system
 - The decimal number system
 - The binary number system
 - Units of data representation
- Coding methods

3. Computer Hardware

- Input unit
- The Central Processing Unit
- Memory
 - Read Only Memory
 - Random Access Memory
- Output unit
- The bus system
- External Memory

4. Computer Software

- Systems software
 - Operating Systems
 - Functions of an operating system

- Types of operating systems
- Examples of operating systems
- Programming Language software
 - Categories of Programming Languages
 - Assemblers, compilers & Interpreters
- Application software
 - Word Processing
 - Spreadsheets
 - Database Management Systems
 - Computer Graphics
 - Integrated Packages

5. Computer Communications

- Local Area Networks (LANs)
- Wide Area Networks (WANs)

2.6 Instructors

The instructors for this course are experts (i.e. software experts, hardware experts, systems experts, and information experts) from the different Departments and Teams of NCIC. The instructors have at least a bachelor's degree in computer, information or related subjects most of them with master's degree in computer and information sciences.

2.7 Problems

The problems identified from formal and informal discussions with instructors, the results of students' assessment of the course, and problems encountered by trainers regarding this course are as follows:

- Students usually come with the expectation of some interaction with the computer in learning a computer course. They are mostly bored with this subject since it is based entirely on lectures and manuals.
- The students are admitted to the course with their varying educational background (i.e., high school completers, bachelor's or master's degree holders, etc.). Therefore, it is not easy to teach students in a uniform pace.
- Students suggest that the medium of instruction to be at least partially in Amaharic. They hint that new concepts are easily understood with a local language than with a foreign language.

CHAPTER THREE

COURSEWARE

3.1 Computers in Education

The main challenge during the first few decades of computing was to invent and improve hardware to reduce the cost of processing and storing data. In the past ten years, advances in micro electronics technologies have successfully achieved more computing power and data storage at increasing lower cost. These rapidly advanced technologies lead to the popular use of microcomputers among all sectors in the society, including the use of microcomputer-based learning in education (Wong, 1993).

The term Computer Aided Instruction (**CAI**) is used to represent the application of computers in education. It is one of a plethora of terms, most with overlapping meanings, related to the use of computers in education; others include **Computer-Augumented Learning (CAL)**, **Computer Based Learning (CBL)**, **Computer Managed Instruction (CMI)**, **Computer Based Training (CBT)**, **Computer Based Education (CBE)**, **Computer Aided Learning (CAL)** **Computer Aided Teaching (CAT)** (Microsoft Corporation 1993,Wong 1993, Creanor and Durmdell 1994, Choo 1994, Alemayehu 1995).

In this study, unless quoting someone, the term Computer Aided Instruction (CAI) is used to refer the use of computers in education.

3.2 Definition

The term courseware was used by Choo (1994) to mean "a teaching software". Hendrickson, Janson, and Rossi (1985) called Computer Aided Instruction (CAI) products as courseware.

Alemayehu (1995) defines courseware as "a system where keyboard, CRT display, other computer controlled equipment (optional) and documental references (Optional) can be used for the purpose of performing explanation by instructor or educational material, response by learners, and further explanation by instructor or educational material" after assuming the following model as the representation of the basic concept of education.

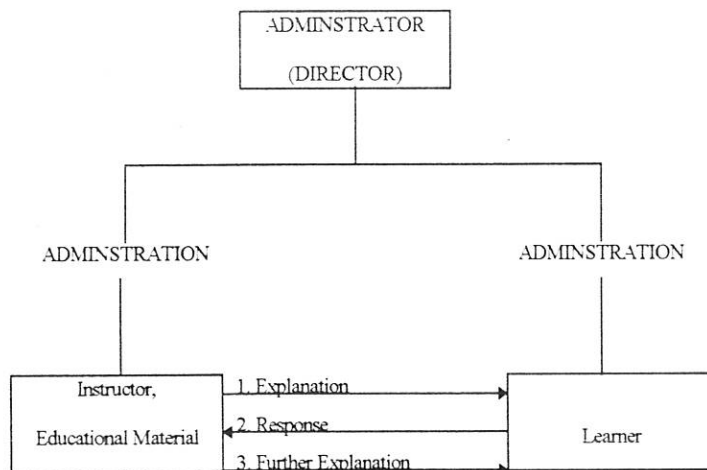


Figure 3.1 Representation of the basic concept of education

Brief review of definitions of courseware given by different authors is given below. Some of them can be found in Alemayehu (1995).

Bunderson (1981) defines courseware in the following way.

"... courseware typically refers to whatever combination of on line and off line materials which comprise a training course or program."

Watts (1981) considers coursewares as

"... those packages of software and ancillary materials specifically designed for classroom instructional use"

Coburn and his associates (1985) refers courseware as

"... educational software which is usually accompanied by a range of ancillary materials including teachers manuals, ditto sheets, work books and tests"

Software packages which help teachers to improve their teaching in the classroom is also referred by Jaques Hebertt as "courseware" (Tagg and Lovis 1988)

In addition Uffing Shouten (Tagg and Lovis 1988) refers courseware to represent the development of CAL as *"... the development of CAL also known as courseware "*. While Radev uses the term to refer to

" ... the preparation of teaching programs or teaching ourses, i.e., coursewares" (Tagg and Lovis 1988)

As can be seen, from the various definitions presented above, although the wordings differ, one can clearly see that the concept emphasized is more or less the same.

3.3 *Advantages of Computer Aided Instruction*

CAI methods provide many opportunities for individualized learning which include: greater motivation, learning privacy, improved information retention, immediate feedback, the opportunity for lessable students to benefit from repeated use, and more active participation by students (Choo 1994).

There is no generally accepted definition of Computer Aided Instruction. It surely implies the use of computers in one or several stages of the educational or training process, but there is no agreement on the scope of its application. Examples range from the most simplistic grade school page-turners and 'drill sargents' to relatively powerful mainframe knowledge-based packages that teach advanced calculus or simulate the re-entry sequence of a spacecraft (Hendrickson, Janson, and Rossi 1985).

Still, almost all Computer Aided Instruction efforts share similar advantages. Their importance varies depending on the approach used, the goals, and the environment in which they are put into practice.

Generally almost all of the applications of courseware are driven by the need to improve the efficiency or effectiveness of teaching, which in turn means getting better instruction results (i.e. improved student achievement or job performance) with fewer resources.

Increasing efficiency/effectiveness benefits of a courseware are more expressed in terms of (Alemayehu 1995 and Hendrickson, Janson, and Rossi 1985):

- increased control;
- reduced resource requirement;
- individualization;
- improved student performance;
- increased learning satisfaction;
- convenience; and
- the power of the computer.

Each of these benefits are briefly described as follows

3.3.1 Increased Control

Lack of control over instructional activities is one of the biggest failings in educational systems. Institutions spend a considerable amount of time and money for developing a course. Such course materials, however, may be modified by instructors which may produce different outcomes than were originally intended. This could lead to lack of standardization. In addition,

except for certain check points, real controlling over the learning process to ensure if students have learned good troubleshooting or diagnostic skills is impossible (Alemayehu 1995).

3.3.2 Reduced Resources

An area where CAI is particularly valuable is in simulations in which students are free to explore the outcomes or consequences of decisions and actions. Simulations are directly applicable to situations where the actual use of complex or sophisticated systems or equipment would be too costly or impossible without the computer. Included are simulations of dangerous or destructive processes, troubleshooting, emergency management and diagnostics. Simulations or "game playing" CAI applications can provide a rich range of experience and give the student the opportunity and ability to learn on the basis of experience (Hendrickson, Janson, and Rossi 1985).

In addition to providing the kind of practice-hands-on-equipment-needed without the cost (risk) associated in the actual equipment, with the advent of computer graphics technology, computer-based simulations are often providing the student with a better understanding of the equipment than is possible with the actual equipment.

Coursewares have also the potential to enable students to take care of the routine aspects of the instructions by themselves through on-line and off-line self study and making the instructor needed only to help students with problems (Alemayehu 1995).

3.3.3 Individualization

One of the central focus of coursewares is allowing each student to learn in his/her own pace and in a fashion most suited to his/her particular learning style (Alemayehu 1995).

CAI products can be designed in such a way that each individual has a unique interactive experience based on their performance. The effectiveness of the learning experience is greatly enhanced if the material sequence is systematically modified to suit each individual's needs. Courseware can review and reinforce material when the student's response is not adequate, and summarize problems or present more advanced material as appropriate depending on his or her learning ability or scope of interest (Hendrickson, Janson, and Rossi 1985).

One advantage of CAI is that it can give the student control over the pace of learning. This control is impossible in class or lecture situations, where it is often difficult for students to interrupt while the session continues at its own pace independent of a student's reaction or understanding of it. In this sense, CAI provides great advantages for both the slow learner and the gifted person (Hendrickson, Janson, and Rossi 1985). It allows students to speed at their own pace, taking time to repeat items if necessary and to practice exercises and examples without the embarrassment of making mistakes in front of colleagues or a teacher. It leads to confidence building through achievement, not only in the specific area but also in the handling of the technology which in itself may be a new skill (Cleanor and Durndell 1994). The value of this characteristic is greatly enhanced if the student has control over both the content and the level of the detail of the learning sequence (Hendrickson, Janson, and Rossi 1985).

Use of courseware allows students not only to proceed at their own pace, but to skip over materials in which they already have competence, and also repeat lessons as necessary. This results in significant time savings over conventional classroom instruction which tend to be geared to the pace of the slowest learner.

Coursewares also allow students to have some control (sometimes free and sometimes conditional) over the order in which they learn topics, (i.e., the sequence of instruction). It is also reported that, the use of courseware enables each student to complete the training with the same acceptable level of competence (Alemayehu 1995).

3.3.4 Improved Student Performance

One of the most widely accepted advantages of CAI is that it makes learning an active process which improves student performance or reduces performance problems (Alemayehu 1995).

Most educational methods (i.e. classes, lecture, readings) allow the student to adopt a passive attitude which result in little actual learning as the student spends most of the time not attending to the instruction. A well designed CAI product forces students to participate actively by answering questions and making decisions (Hendrickson, Janson, and Rossi 1985).

Courseware's capability to provide interactive and individualized instruction provide it with inherent potential to improve the quality of teaching (Alemayehu 1995). Most psychologists agree that active involvement facilitates learning, even more when students are forced to play a constant thinking role and the experience emulates a conversation between two people

(Hendrickson, Janson, and Rossi 1985). Such continuous interaction and two way communication, according to Alemayehu (1995) and Hendrickson, Janson, and Rossi (1985), results in actual learning and improved student performance.

3.3.5 Increased Learner Satisfaction

The interactive nature of courseware has a common outcome of improved learner satisfaction and enjoyment. Students typically find using courseware more motivating than other forms of instruction (Alemayehu 1995).

The feedback provided on responses and the capability of assessing the student progress available in most of the coursewares creates an environment whereby students develop a sense of achievement while learning. This leads to increased satisfaction during training which is one of the most important factors that contributes to students achievement and satisfactory completion of the course (Barker and Manji 1992). Poor motivation rather than lack of ability or skill often causes learning problems. Thus, the capability of coursewares to improve motivation is an important factor in reducing course failures.

3.3.6 Convenience

Considering some of the reasons discussed above, convenience may appear to be a relatively weak rationale for use of coursewares. However, as computer systems become more ubiquitous and widespread, it is becoming a very strong and compelling argument for courseware.

In so far as students have access to a computer system when they go to work, which is most likely to be the case, and to a courseware package, use of courseware can make a good deal of sense. In addition, the courseware can always be used for refreshing previous knowledge.

3.3.7 Power of the Computer

Another virtue of the use of computers in education is the flexibility and power of the computer itself. The computational ability and capacity to drive other equipment permits integration processes. Slide projectors, laboratory equipment, tape recorders, video tapes and laser disks are examples of equipment which have been used for some time to enhance the learning process, but results from their individual use have been mixed. Modern computers can now either simulate their performance (through improved graphics and sound generation technologies) or couple their capabilities as global monitors (Hendrickson, Janson, and Rossi 1985).

More compact storage of information is also an advantage of CAI. While computers themselves require considerable space, it is possible to pack a considerable amount of written material and instructions on a small disc or chip. Advances such as laser discs promise considerable improvement in this regard.

Finally, the reliability of machine operation is an advantage, as Main (quoted in Hendrickson, Janson, and Rossi 1985) said "the systems are patient, never have a day off, don't loose their tempers, and have all the time in the world".

While these are some of the main arguments generally cited to encourage the use of courseware, increased development time, problem of facilitating man machine communication, problem of cost efficiency and that of user acceptance are, however, the main reasons cited to discourage its use (Hendrickson, Janson, and Rossi 1985).

3.4 Types of Courseware

There are many different types of Courseware, each one having different characteristics. The four most common are:

1. **Computerized Tutorials.** This is one of the most comprehensive forms of application of computers in education. Tutorial coursewares involve the presentation of information. Such coursewares typically consist of discussions of concepts or procedures with interspersed questions or a quiz at the end of the discussions. The primary purpose of tutorial applications is to teach new information. In this regard, tutorial coursewares, by taking the full instructional burden, guides a student to the achievement of a specified set of objectives (Alemayehu 1995).

The first implementations of tutorials were simply "books on a screen". The first and most common form was linear, where the student was presented with a sequence of displays consisting of text and graphics which defined the lesson and a group of questions and feedback responses that would evaluate the student's understanding of the topic. With time, the learning process was enhanced by using the student's responses to

decide whether to repeat the material, allow him to branch to the next sections of the tutorial, or provide him with additional information (Hendrickson, Janson, and Rossi 1985, Alemayehu 1995).

The critics of the tutorial coursewares point out its "limited and limiting pedagogy" (Alemayehu 1995). They are also criticized by their nature of trivializing concepts, i.e., by narrowing the range of possible responses, tutorial coursewares keep students from exploring the complexities of a given concept.

Advocates of tutorial courseware, however, respond to this criticism by pointing out that the same criticism can be applied in the conventional instructional methods such as the lecture, textbooks etc. (Alemayehu 1995). With the advent of intelligent tutorials which make the representation of knowledge very possible and an easy task, such criticism is losing its significance.

2. **Computerized Drill-and-Practice Sessions.** The purpose of this type of CAI is to provide feedback and practice on skills already learned (as compared with tutorials which primarily teach new skills). This is the most common type of CAI now used. It generally is a two-phase process where the student is asked a series of questions and provided with feedback about the correctness of his answers or diagnostics of his mistakes. Drill and practice lessons are a natural follow-up for tutorials. Drill and practice CAI may be quite sophisticated, including "intelligent" expert system based tutors (Hendrickson, janson, and Rossi 1985, Alemayehu 1995).

The critics of this type of courseware include pedagogical narrowness (lack of stimulus - response) and the possibility of reinforcing incorrect learning. But advocates of drill and practice type of courseware acknowledge that such criticism is due to poor design of the programs by the persons (educators) involved in the development rather than the inherent nature of drill and practice. They suggest that with the additional features of graphics, sound and animation drill and practice can provide useful educational experiences (Alemayehu 1995).

3. **Computerized Simulations.** Simulations typically involve the specification of an objective and the presentation of information about the current state of the system to be studied. The goal is to teach the student to identify and control the relationships between different components of the system while trying to reach a certain goal. In the process, the student is provided with feedback regarding the consequences of his actions or responses (Hendrickson, Janson, and Rossi 1985).

Simulations are based on a model of some process, mechanism or activity. Such model, allows a student to relate input or changing parameters to output or outcomes; initiates a real or imaginary system based on the model's theory of the operation of the system. Since reality cannot be fully represented by any model, simulations focus student's attention on certain aspects of the process under investigation. Simulations, however, are also powerful learning tools in studying events that could not be otherwise examined owing to danger, expense, or lack of time (Alemayehu 1995).

Although simulations have such a profound use, the design and programming of good computer simulation is very difficult. The main critics in this line is the inherent difficulty that exist in representing real world problems with the model (Alemayehu 1995).

4. **Instructional Games.** Instructional games involve an element of fantasy which is not possible in the other types of coursewares. They are designed to be fun for students and thereby increase the chances of the student learning the concept, knowledge or skill embedded in the game. Game coursewares are most helpful and effective in a situation where student motivation is a concern (Alemayehu 1995).

The most successful instructional computer games have challenge, fantasy and curiosity characteristics. To provide such features, the development of instructional game coursewares require sophistication in both curriculum design and computer programming.

Besides using a different approach and methodology, each is used at different stages of the learning process. The tutorials tend to replace formal lecture education, drill-and-practice tend to complement and develop skills already learned, while the simulation-game is generally a final step towards integration and effective use of skills in problem solving. Thus, tutorials, drill and practice sessions, simulation, and gaming each play an important role in the educational process that leads from general principles to their actual use in pseudo-real situations (Hendrickson, Janson, and Rossi 1985).

3.5 Courseware Design Characteristics

An increasing number of educational software have been marketed by commercial firms and individuals all over the world.

In order to evaluate and choose among these coursewares and/or develop a courseware to pass the evaluation, identification of the basic features that should be possessed by a courseware is essential.

Alemayehu (1995) provides basic characteristic features that should be present in a particular courseware. He has categorized the different characteristic features of a courseware into four: factors related to **program content, pedagogy, program operation, and program documentation.**

The **PROGRAM CONTENT** characteristic feature of a courseware include:

- presentation of information about the content and background;
- suitability of the content of the material for students' use;
- compatibility of the content with the learning outcome(s) expected;
- educational significance of the content;
- statement of objectives and aims;
- statement of the intended type of use and audience; and
- availability of instructions for running the program.

The **PEDAGOGY (FRIENDLINESS AND FLEXIBILITY OF THE PROGRAM)**

characteristics are:

- provision of immediate feedback which may be passive, active or interactive;
- clear definition of the learning theory underlying the courseware, i.e., behavioral, cognitive, logical, etc. ;
- self contenance of the program, i.e., allowing students to learn by their own without intervention when needed;
- effective screen presentation, i.e., usage of color, sound, graphics, and animation to enhance student learning without filling the program with unnecessary sound and fury; and
- motivation (motivating) of students in using the program.

The **PROGRAM OPERATION** characteristics include:

- clarity of the program from bugs and breaks;
- clarity and acceptance of the directions and instructions;
- synchronized usage of graphics, sound, and color capabilities;
- informative and clear user error handling system;
- versatility, i.e., allowing the student control (conditional or unconditional) over the program operation; and
- portability, that is ability to transfer the program to a different computer.

The **DOCUMENTATION** characteristics of a courseware are:

- information about machine requirements;
- information about the model used;
- information about the program structure;
- listing of the program code; and
- user documentation to use the package.

Generally if one expects to gain the benefits of using a courseware, the courseware has to satisfy all or most of the above characteristic features.

3.6 Current Trends and Developments

Within any area of endeavor, developments in technology usually affect progress in four important ways. First, technology can be used to extend the limits of what can be done; second, it can affect the speed with which objectives can be realized; third, technology can strongly influence the ease with which activities can be performed; and finally, technology can be used to extend the area of endeavor in new directions as a result of providing 'enabling resources' (Barker 1993). Owing to the evolution of technologies, courseware development has made appreciable progress in recent years. In this regard, the application of technologies like object oriented programming, hypertext, artificial intelligence (particularly expert systems) efficiently contribute to the design and pedagogical maintenance of interactive and intelligent coursewares.

A brief description of how these technologies contribute to the design of courseware is presented as follows.

3.6.1 Hypermedia

One of the main obstacles that holds back the production and diffusion of educational courseware is the excessive amount of time necessary to develop a quality courseware. The hypermedia approach provides a solution to both the problem of quality and development time (Alemayehu 1995).

Hypertext refers to a non-sequential, non-linear method for organizing and displaying text (Jonassen and Wang 1993). When these data are graphics or sound as well as text or numbers, the resulting structure is referred to as hypermedia (Parsaye et al. 1989). A Hypermedia environment is the user interface that allows to work with different elements as a whole: text, graphics, images, sounds, simulation software, external hardware, databases, computer networks, etc., on the basis of specific application. It is designed to enable the reader to access information from a text in ways that are most meaningful, based upon the assumption that the organization that the reader imposes on a text is more meaningful than that preferred by the author.

The most pervasive characteristic of hypertext is the node, which consists of chunks or fragments of text, graphics, video or other information. The size and modality of a node varies from a single picture or a few words to the size of a large document. Nodes are the basic unit of

information storage in a hypertext. Modularizing information enables the user of the information system to determine what chunk of information to access next. Another important characteristic of nodes in many hypertext systems is that they are amendable or modifiable by the user. The user may add to or change the information in a node or create his or her own nodes of information.

The organization of a hypertext, that is, the interrelationships between the nodes, is defined by the links that interconnect the nodes. Links in hypertext systems are typically associative, that is, they define an associative relationship between the node pairs that they connect. The links transport the user through the information space to the nodes that are selected, enabling the user to navigate through the knowledge base. The node structure and the link structure form a network of ideas in the knowledge base, the interrelated and interconnected group or system of ideas.

Hypermedia permits the user to determine the sequence in which to access information (browsing), to add or amend the information in order to make it more personally meaningful (collaboration), or to build and structure their own knowledge base. Like most information systems, interaction is one of the most important attributes. Hypertext information systems permit more dynamic user control of the information in the knowledge base (Jonassen and Wang 1993).

Any piece of information be it text, graphics, sound or numerical data, can be linked to any other piece of information. This flexibility makes it possible to construct a qualitative courseware

information storage in a hypertext. Modularizing information enables the user of the information system to determine what chunk of information to access next. Another important characteristic of nodes in many hypertext systems is that they are amendable or modifiable by the user. The user may add to or change the information in a node or create his or her own nodes of information.

The organization of a hypertext, that is, the interrelationships between the nodes, is defined by the links that interconnect the nodes. Links in hypertext systems are typically associative, that is, they define an associative relationship between the node pairs that they connect. The links transport the user through the information space to the nodes that are selected, enabling the user to navigate through the knowledge base. The node structure and the link structure form a network of ideas in the knowledge base, the interrelated and interconnected group or system of ideas.

Hypermedia permits the user to determine the sequence in which to access information (browsing), to add or amend the information in order to make it more personally meaningful (collaboration), or to build and structure their own knowledge base. Like most information systems, interaction is one of the most important attributes. Hypertext information systems permit more dynamic user control of the information in the knowledge base (Jonassen and Wang 1993).

Any piece of information be it text, graphics, sound or numerical data, can be linked to any other piece of information. This flexibility makes it possible to construct a qualitative courseware

because one is able to represent knowledge; browse, carry out structured searches and make inferences all within the same environment (Parodi 1993).

Hypertext also enhances co-operative writing among multiple authors. It allows one not to commit him/her self to a particular linear sequence of ideas when writing. Ideas can grow and evolve naturally over time while a consensus built on how they should be linked. It enables to avoid most disputes over material organization and primacy by allowing variety of orders to be constructed and tested.

Generally, the realization of multimedia application coupled with usage of an object oriented environment avoids the need for writing a long list of program statements. Instead it is possible to build applications by creating and/or modifying the objects that embody the different parts and functions for those applications. The hypermedia environment provides a facility for developing a courseware which can facilitate individualization and interaction (Alemayehu 1995).

In this research hypertext is a matter of concern.

3.6.2 Expert Systems

The subject of Artificial Intelligence (AI) provides the most promising source of new ideas for courseware. The particular branch of Artificial Intelligence which is currently making a significant impact in education relates to Expert Systems.

Expert systems are intended to capture and make accessible the operational knowledge of an expert. An expert system is an intelligent computer program that uses knowledge and inferential procedures to solve problems that are difficult enough to require significant human expertise for the solution. Expert systems seek to use those procedures and inferences to model the thought processes of an expert practitioner. The expert possesses the knowledge (facts, principles, strategies) that enables him or her to make decisions and solve problems. In order to be useful for building expert system knowledge bases, this knowledge needs to be translated into facts and heuristics.

There are several classes of tasks for which expert systems can function most effectively, such as (Jonassen and Wang 1993):

- automating procedures,
- diagnosing problems (such as medical),
- monitoring any activity or process (such as patient care).
- configuring new systems (such as a new computer system),
- scheduling operations (such as a major league baseball schedule), and
- planning or managing large processes (such as constructing a mall).

Expert systems provide the tools for knowledge representation which are referred to as knowledge base, inferencing and higher level programming capabilities. Expert system environment also includes a number of tools for helping the various peoples who build or use the expert system.

The knowledge base is the part of the expert system that carries out long term memory of facts, structures and rules that represent expert knowledge about the domain of experts. Whereas the inference engine is the part of the expert system that carries out the reasoning function.

In courseware development it is not the expertise of an expert system that is used to make improvement but the usage of expert systems as a higher level programming tools. In particular it is the approach to expert system where they are seen primarily as high-level programming environment for information and knowledge-intensive tasks which dominates the advantage of expert systems in courseware design and development. That is, expert systems technology are used as a general programming environment for putting intelligence into coursewares.

Such inclusion and representation of knowledge using the expert systems results in the development of intelligent coursewares which are highly interactive and have the capability of modelling the students knowledge in order to discover the reason(s) of his/her mistakes. As such expert system application increases the adoption of the courseware to a given student. It also facilitate development of courseware packages which can respond flexibly to individual learner needs and progress (Alemayehu 1995).

3.6.3 Expertext

The term 'expertext' was used by Rada to describe the work he has undertaken in the context of developing 'intelligent hypertext' by augmenting hypertext through the use of expert systems technology (Barker 1993).

Jonassen, Grabinger, and Wang (1993) demonstrated different integration of hypertext and expert systems. Hypertext can be used as a front-end interface and back-end help system for expert systems. Expert systems can function as diagnostic tools for accessing information in a hypertext knowledge base, for describing the expert's knowledge base in order to prescribe the most productive learning paths, or as a means for modeling student learning in intelligent tutorials (Jonassen and Wang 1993).

Obviously, in order to undertake developments of this sort it is important to have suitable development tools available. One very useful tool that combines a hypermedia authoring facility and an expert system capability is Knowledge Pro system (Barker 1993). This development tool provides a very powerful environment with in which to produce different types of intelligent hypermedia system that incorporate appropriate combination of hypertext, sound and pictures (Barker 1993).

Jonassen and Wang (1993) has demonstrated the use of the integrated Hypertext/Expert system (Expertext) by developing a prototype intelligent tutoring system (ITS) using hypermedia and

expert systems called the Physics Tutor, an instructional physics information base designed to familiarize students with the basic concepts in rectilinear Newtonian Physics (Jonassen and Wang 1993).

3.6.4 Object Oriented Programming

One of the promising approaches to the courseware development challenges is object oriented programming. This technology for software development enables the programmer to arrange groups of code (objects) to build an application rather than building each application from scratch.

Systems built around object orientation methodology are built as collections of classes whereby every class represents a particular abstract data type implementation or group of implementations.

Generally, object orientation can be characterized through the most fundamental aspect of its paradigm: abstract data typing (encapsulation), inheritance and object identity (Parodi and Ponta 1993 and Parsaye et al. 1989)

The first concept, abstract data typing, refers to the attributes and behaviour of an object. It enables to keep critical information together, hides details, minimizes dependency and allows new objects to be easily added. The concept of inheritance refers to the nature of all the objects responding consistently to the same set of commands that are inherited from the original objects.

The third concept, i.e., identity allows easier maintenance by packaging different objects with the same commands which enables to respond to a consistent message.

Object oriented programming, generally, allows a more direct representation of the real world model in the code. As a result, the normal radical transformation from systems requirements (defined in user's term) to system specification (defined in computer term) is greatly reduced. It provides better concepts and tools to model and represent the real world as closely as possible.

Application of object orientation is increasingly becoming popular in software engineering. In particular, object oriented programming is found to be suitable for meeting such software engineering goals as modifiability, maintainability, understandability, portability, abstraction, information hiding, reusability etc. (Parsaye et al. 1989, Scholtz et al. 1993 and Arano et al. 1993).

The main features of the object oriented model mentioned earlier, i.e., encapsulation, identity and inheritance provide the facility to design software which could be easily modified, maintained, understood and portable without any effect on the rest of the system.

Courseware development (engineering) basically being an activity of software engineering, which is differentiated from the later only by its educational content, draws all the benefits of using object orientation for software engineering. Especially, Parodi and Ponta (1993) have reported that, the application of object orientation methodology in courseware development enables an

incremental development and software reuse. It also efficiently contributes to the design and pedagogic maintenance of interactive coursewares.

The implementation of object oriented languages enables the integration of all the courseware features into a homogeneous parts where all the different features of a lesson are built as "objects" and as such can be efficiently manipulated and combined. In fact, instead of building each application from the beginning, it enables the courseware programmer to accomplish this task by the arrangement objects.

Generally Cox and Hunt point out that "... what is truly revolutionary about object oriented programming is that it helps programmers reuse existing code, just as Silicon Chip help circuit builders reuse the work of chip designers" (Parsaye et al. 1989).

CHAPTER FOUR

HYPertext

4.1 Introduction

Information (and knowledge) can be organized in a variety of different ways. Naturally, the way in which these commodities are organized and stored can significantly influence their meaning and the purposes for which they can be used. Two basic approaches are often used for organizing (and presenting) information and knowledge, often referred to as the linear and the non-linear approaches.

4.1.1 Linear Structures

The essential idea underlying a linear organization is that a corpus of knowledge is organized into several units which are designed to follow each other in sequence, although some jumping back or skipping ahead is usually allowed. Each of the basic units from which a linear structure is composed has just one entry point and one exit. Another important point to note is that the material embedded within each module that makes up a linear information structure is intended to be processed in a strictly sequential fashion (Barker 1993).

The linear approach has, until recently, been one of the most widely used types of organizational structure. Indeed, it forms the basis of many of the most commonly used methods of organizing and presenting knowledge. Consider, for example, the collection of chapters from which a book

is composed - they are arranged into a particular sequence in order to achieve a certain effect. Each chapter is composed of a number of sections each of which is made up from a series of paragraphs. Paragraphs are composed of sentences which, in turn, are built from sequences of words. Although readers may not choose to access the book in this fashion, it is certainly the way in which the author intended it to be used at least on first reading (Seyer 1991, Barker 1993).

Conventional books thus provide a good example of an essentially linear information storage and presentation mechanism. Many other types of media also impose a linear structure onto the information they store. Two common examples are audio tape and film. Each of these storage media is intended to store information units (and subsequently present them) in a strictly sequential fashion (Seyer 1991, Barker 1993).

4.1.2 Non-linear Structures

Some information storage and presentation media (particularly those which embed some form of computer facility) allow significant departures from the basic linear strategies described above. In a non-linear structure the basic units of information that make up the knowledge corpus are joined together in many complex ways that allow them to be processed in a variety of different non-linear pathways. This is made possible because each of the modules used in this type of arrangement has one entry point but can have any number of exit points. Therefore, from within the body of one unit it is possible for the user to 'jump' to any of a number of other units depending upon his or her information interests and requirements (Seyer 1991, Barker 1993).

The essential feature of a nonlinear information/knowledge unit is the existence of embedded 'decision points' at which the user has to make a decision about which way to proceed. For example, at a certain decision point, the user can choose to exit or carry on within the same unit; at another decision point a similar choice is available - carry on or leave the unit via any of the exits in that unit (Barker 1993).

Each of the exits from an information unit will usually lead to a different outcome. Also, when a given exit is chosen the user may leave the current 'host' information unit permanently, temporarily or transiently, depending upon the effects that the designer wants to achieve (Barker 1993).

The decision-making inherent in using a non-linear information structure is performed interactively by the user. Usually, decisions are made using the 'point and click' facility provided by the pointing device that is attached to the delivery platform being used to present the information (Seyer 1991, Barker 1993).

The name hypertext is given to a text structure of the above form. That is hypertext is nonlinear, or non sequential text. The text is organized so you can easily jump around from topic to topic. You do not need to read the text in a fixed sequence (Seyer 1991).

In a typical hypertext application, users enter the database by accessing a table of contents, searching through the index for an entry point, or starting to read from a designated beginning. As users read through the text, various words or graphics appear as buttons-that is highlighted or otherwise visibly connected to links with other material. By pointing with the cursor and either clicking a mouse or pressing a key, users move instantly to other material by way of predetermined links (Carl Binder 1992).

4.2 Definition

The word 'hyper' in hypertext has the meaning of 'extending in to another dimension' as in hyperspace, hypersphere, or hyperdimensional (Seyer 1991, Corenjo 1994).

In its literal sense, the term hypertext implies extra dimensions to text. In practice the term is often applied to computer system which allow a person to browse a document by deliberately jumping from text block to text block (Rada 1991). According to Parsaye et. al. (1989), hypertext is a tool for building and using associative structures. While a normal document is linear and requires reading it from the beginning to the end, reading hypertext is open-ended and one can jump from one point to the other depending on the interests. The nearest thing to a hypertext that most people are familiar with is a thesaurus. A thesaurus is not normally read from the beginning to the end. Each time a thesaurus is consulted, it is entered at a different location based on the word used to initiate the search, and once the sought term is located, there are pointers which lead the user to other parts of the thesaurus to get more information on the terms related to the sought term. Hypertext can be thought of as an enriched thesaurus where, instead of links between words, links between documents and text fragments are available (Parsaye 1989).

According to Barker (1993) the term hypertext and hypermedia were first introduced by Theodore Nelson who defined hypertext as being 'computer supported non-sequential writing'.

Slatin (1991) defines hypertext as an assemblage of text, images, and sounds - nodes - connected by electronic links so as to form a system whose existence is contingent upon the computer. The

user/reader moves from node to node either by following established links or by creating new ones.

According to Lucarell and Zanzi (1993):

Hypertext can simply be defined as a system to manage a collection of information that can be accessed non-sequentially. It consists of a network of nodes and logical links between nodes. The variety of nodes and links that can be defined make hypertext a very flexible structure in which information is provided both by what is stored in each node and by the way the information nodes are linked to each other.

According to Microsoft Corporation (1993), Hypertext is a metaphor for presenting information in which text, images, sounds, and actions become linked together in a complex, nonsequential web of associations that permit the user to browse through related topics, regardless of the presented order of the topics.

Hypertext, at its most basic level, is a DBMS that lets one connect screens of information using associative links. At its most sophisticated levels, hypertext is a software environment for collaborative work, communication, and knowledge acquisition (Fiderio 1988).

Hypertext is, according to Binder (1992), a network of cross references between related pieces of information that need not bear any formal similarity.

Hypertext can be defined as a computer-based system which allows users to access related information in various forms by means of activating links which in turn lead to the display of new data (Creanor and Durndell 1994).

Hypertext is a text database + semantic net + interface. It is a system having three components (Rada 1989, Frisse and Cousins 1992):

1. a database of text
2. a semantic net which connects the text components, and
3. tools for creating and browsing this combination of text and semantic net.

Hypertext is a non-sequential, non linear method for organizing and displaying text (Jonassen and Wang 1993).

4.3 Historical Background

Although Ted Nelson coined the term "hypertext", in 1965 Vannevar Bush was probably the first to discuss the concept in print 20 years earlier.

Over the last few decades considerable attention has been devoted to research and development of hypertext (Berk and Delvin 1991, Seyer 1991, and Barker 1993). In the following table we shall see a hypertext timeline (Berk and Delvin 1991).

Table 4.1: Hypertext timeline

Year	Event
1945	Vannevar Bush published an article entitled "As We May Think" in the Atlantic Monthly in which he claimed that the progress of research was being stymied by the inability of researchers to find and access relevant information. Bush proposed the "memex" system, a microfiche-based system of documents and links which foreshadowed the advent of hypertext. Some of the requirements for the memex as specified by Bush were: fast access to information, ability to annotate, and the ability to link and to store a trail of links.
1962	Douglas Engelbart published a paper, 'Augmenting Human Intellect: A Conceptual Framework,' which set Engelbart's agenda for the 30 years that followed. Engelbart sought to define and implement the functionality necessary for computers to augment human abilities. The functions he thought necessary included links between texts, electronic mail, document libraries as well as separate "private" space on the computer for users' personal files, computer screens with multiple windows, and the facilitation of work done in collaboration by more than one person. In the course of his career, Engelbart invented the mouse, outliner and idea processor, and on-line help systems integrated with software. Engelbart was responsible for the first substantive implementations of electronic mail, word processing, and shared screen teleconferencing.

1965 Theodor Holme Nelson invented the term "hypertext" and presented it to the world. He constructed Xanadu, a hypertext engine based on his version of hypertext.

1968 Andries Van Dam and his team at Brown University developed the Hypertext Editing System, which was intended to serve two purposes: to produce printed documents nicely and efficiently and to explore the hypertext concept. Van Dam's second hypertext project at Brown was called the File Retrieval and Editing System (FRESS, completed in 1982). According to Van Dam, FRESS was the first system to have an undo function; undo remains the most popular feature of many software packages.

The latest hypertext project at Brown is called Intermedia. Intermedia-based applications are used in teaching and learning of biology and English Literature at Brown. Intermedia is used both as a tool for professors preparing their lessons and course materials and by students for learning and creating reports. One of the seminal ideas derived from Van Dam's work is that of the "web", a set of links that belong together. By opening a web, a hypertext reader imposes a particular set of anchors and links on a document. This makes it possible for different users to impose their own sets of links on the same document.

1972 The last of the "first generation" hypertext systems, ZOG, was developed at Carnegie-Mellon University. The first generation systems all originally ran on mainframes, used text to the exclusion of other media, and provided support for

workers to collaborate on a hypermedia network. ZOG was specifically designed to provide fast response to a large number of users.

1983 The "second generation" of hypertext authoring products began in the early 1980s with the emergence of workstation-based, research-oriented systems such as Intermedia and KMS. The difference between the first and second generation of hypertext products had a lot to do with improvements in technology that became available on workstations. These faster computers and displays supported more sophisticated user interfaces than earlier systems could. The second generation systems are generally targeted at networked or UNIX-based, single-user workstations, not mainframes as the earlier products were. KMS, a commercial implementation of ZOG, has been marketed since 1983. KMS is capable of storing text and graphics in its nodes, which are called "frames". It is particularly appropriate for industrial-strength hypertexts, where many designers and engineers must share the same documents on a large computer network.

1985 In 1982, Peter Brown began to invent the first commercial hypertext authoring system for a personal computer, which was called Guide in 1985 when Office Workstations Limited (OWL) began to market it for the Apple Macintosh. Guide was released for use on an IBM PC in 1987. Like other microcomputer-based hypertext authoring systems that came later, Guide provides less functionality than earlier mainframe and workstation-based products, but it does rely on a graphical user interface.

1986 Xerox PARC's NoteCards was released. NoteCards, KMS, and Intermedia all support graphics and animation nodes as well as formatted text. They also all provide graphical overviews of the structure of the hyperdocument to aid navigational access. NoteCards pioneered in the application of metaphor to hypertext; each node in NoteCards is represented on screen as a card. NoteCards can be of any length necessary.

1987 HyperCard came out. HyperCard does not allow text-to-text links. It provides few of the frills most hypertext authors require for large scale hypertext production, but because it comes with every Macintosh sold, it certainly has brought a semblance of hypertext to the Macintosh masses. As can be surmised from its name, HyperCard imposes a card metaphor on its nodes. No node can be larger than a card in size. A hyperdocument is called a stack in HyperCard, and HyperCard programs are called stackware. Cards in HyperCard are linked to other cards via link anchors called "buttons". Also in 1987, the IBM-PC version of Hyperties, which began life in 1983 as The Interactive Encyclopedia System (TIES) at the University of Maryland's Human-Computer Interaction Laboratory, was introduced.

1990 There are now plenty of commercially available hypertext authoring systems running on a variety of platforms. New versions of existing hypertext products such as HyperCard, Hyperties and Guide are released regularly. Hypertext conversion programs such as Texas Instrument's HyperTRANS, OWL's IDEX, and Big Science's SmarText (now owned by Lotus), which convert existing

electronic documents into hypertext, are becoming viable options or additions to business documentation plans.

4.4 Types of Hypertexts

The most pervasive characteristic of hypertext is the node, which consists of chunks or fragments of text, graphics, video or other information (Jonassen and Wang 1993).

A node is a collection of data organized around a specific topic. In different hypertext systems, different terms are used for nodes. In hypercard, a card is a node. In HyperPad, the node is a pad. HyperWriter and xText talk about nodes as pages (Seyer 1991).

In the early hypertext systems nodes were always text nodes. The availability of computer peripherals to generate sound and to deal with visuals creates a new communication opportunities. A node might be (Seyer 1991):

- text display;
- graphic display;
- video or a video disc segment;
- animated graphics;
- recorded speech;
- digitized speech; and
- music;

If the word "text" in the definition of hypertext is replaced by "music", "film", "images", "sound" and so on, a hypermusic, hyperfilm, hyperimages, hypersound systems, etc. could be defined (Rada 1989, Barker 1993). The term hypermedia is used by some people when referring to systems that include a wide variety of node types (Seyer 1991).

Since Ted Nelson first coined the term "hypertext", there has been debate over whether media other than text are encompassed by that term. Nelson himself has acknowledge the usefulness of the word "hypermedia", although he claims that from the first, his vision of hypertext always embraced multimedia as well (Berk and Devlin 1991).

In this study the term hypertext is used on its own and as a synonym to hypermedia. To avoid confusion, terms such as text only hypertext and multimedia hypertext are used.

4.5 Authoring Tools

The process of creating a hypertext or hypermedia system is usually undertaken using a suitable authoring system such as GUIDE, HyperCard, HYPERTIES, ToolBook, KnowledgePro, and so on.

The power of an authoring environment will depend upon the quality and range of tools that it makes available. Usually, as a basic minimum, an authoring environment should provide tools for (Barker 1993, Perez 1991):

1. creating multimedia information units;
2. marking reactive areas within these units;

3. linking reactive areas within one source unit to appropriate target units in order to create a network structure;
4. importing both marked (using a mark-up language) and unmarked information units for incorporation into the hypermedia network that is being built;
5. updating the network after it becomes operational; and
6. controlling the 'access rights' of the users of the hypermedia corpus (that is, controlling who can do what with respect to viewing, changing and deleting stored information).

4.5.1 KnowledgePro

KnowledgePro is a powerful expert system shell. It has a full-blown language especially designed for creating expert systems with hypertext capability. Its built-in language has about 120 commands. It includes commands for list manipulation, string operations, screen and file input/output, debugging, and logic and arithmetic operation. In addition it has some special functions for returning values from what are called "topics". A topic in KnowledgePro is a kind of amorphous object that can behave in different ways. A knowledgePro topic can (Seyer 1991, Knowledge Garden Inc. 1991):

- Hold commands, like procedures
- Store values, like variables
- Return Values
- Inherit Values
- Be linked into a hypertext network

In addition to being a powerful expert system shell incorporating a hypertext development system, this authoring tool is used for the thesis primarily because it is the only authoring tool available at SISA.

4.6 Educational Application

The application of computers to the process of teaching and learning holds many promises. Their use has been seen as a way of improving the quality of delivery, increase the individualization of learning, encouraging learner autonomy, and, amongst others, providing a cost effective solution to the issue of increasing number of students, particularly in higher education (Barker 1993).

Of all the possible applications of computer technology to learning, hypermedia seems to be one with a great potential. It is non-linear, learning is very often non-linear; it involves the presentation of material using a variety of media, most quality learning uses a variety of media; it allows flexible and differing uses by the learner, learning is optimized when the learner can attain the system to his or her needs and preferred learning styles; it facilitates feedback to the learner, quality learning requires quality feedback (Barker 1993).

Educators are making good use of hypertext and making learning difficult subjects more and more accessible. For example, someone reading instructional material on a computer screen. Several technical terms on the screen are highlighted - shown in a different color. He points to one of the highlighted terms and clicks the mouse button, a new window of information - a definition or elaboration of the term will appear (Seyer 1991).

Yang and Moore (1995) list the following five points as the potentials of hypermedia in education:

1. Providing rich and realistic contexts for multichannel learning: Manipulating sophisticated audio-visual devices, a hypermedia system can present vivid information in various formats. A learner can then select the most comfortable and effective format to grasp the meaning. The huge information base forms a rich and realistic resource for learners to explore.
2. Accessing information non-linearly: Hypermedia breaks the classroom's arbitrary, linear teaching style. It allows learners to access randomly the stored information based on their needs, interests, or whims. The process of finding and organizing information will not only give true meaning to knowledge, but also facilitate the construction of learner cognition.
3. Focusing learner attention of the relationship of facts: The usefulness of facts is limited if they are isolated. The structure of hypermedia systems inherently helps learners focus their attention of learning relationships among facts. Therefore, it produces better usage of information.
4. Encouraging active, student-centered learning: A hypermedia system is an interactive environment. Since there are always decisions about where to link, learners must be actively engaged in the retrieving process. They control learning speed, amount, and path based on their abilities and needs. Hypermedia systems thus provide and individualize environment for learners.

5. Promoting collaboration: Hypermedia systems allow learners to "author" the contents or the structures. Learners annotate, delete, or modify the stored information to correspond to their cognition. Thus, they are not only knowledge receivers. They are knowledge contributors, too. A group of learners can also share a hypermedia package. They work together and exchange ideas through authoring functions. The 'collaborating' process blurs the distinction of authors and users. It also induces the cooperation between learners.

Several problems are identified to plague in the use of hypertext for instruction. Yang and Moore (1995) identified the following problems.

Too much Learner Control:- In hypertext systems, learners are granted full control of navigation without specifying learning objectives or pedagogic strategies. Studies shows that when novices in the subject matter, poorly-motivated learners, or learners lacking meta-cognitive skills take control of learning, better learning effects are not found.

Lost in Navigation:- A hypertext system contains huge amount of information nodes. Most nodes connect with multiple other nodes. While learners have the freedom to jump from one node to another node, they are easily lost in the complex hyperspace. They will not know where they are, where to go, or how to get there. This is specially true for young and novice learners who are easily distracted by unrelated information and then are buried in the fascinating but confusing information maze.

Cognitive overload:- During the navigation in hypertext system, learners have to continuously make decisions about where to go and to make sense out of the connection. There are tremendous efforts demanded in the process. It consumes learners' limited mental resources which otherwise can be used on learning content knowledge.

Missing important information: Most hypertext systems are open environments where no specific objectives or learning paths are specified. It is up to the learner to decide what to learn and when to terminate the learning. The networked paths in hypertext systems are so complex that most learners are unsure of whether or not they have completed the important topics. Also, there is no way to measure whether the learning objectives are satisfied because no summative evaluation or feedback is provided.

To solve those problems, various designers proposed strategies to improve the effects of hypertext systems on instruction. If hypertext systems are going to have significant role in instruction, they must function in as many elements as possible in the instructional process. Therefore, hypertext systems must be designed systematically. In this study the hypertext courseware has been designed to overcome these problems as possible.

4.7 Current Trends and Developments

Hypertext applications are being improved and expanded through the improvement in different technologies. Some of these technologies are briefly reviewed below.

4.7.1 Pointing Devices

Hypermedia is essentially an 'interactive' technology; that is, users point (either directly or indirectly) to reactive objects that are being displayed on a computer screen. These pointing operations enable users to select the particular items of information that are of interest to them. The successful implementation of pointing operations depends upon the availability of a suitable interaction technology, for example, a touch-screen or light pen (for direct pointing) or a mouse, roller controller, joy-stick or tracker-ball (for indirect pointing).

The use of pointing devices similar to those listed above is now well established. However, other types of technology to facilitate pointing operations are being developed. We shall look into three of them.

Data gloves. The data glove is a glove-like device that contains embedded position and direction sensors that enable a computer to deduce the screen location at which a user is pointing with his or her finger (Seyer 1991, Barker 1993).

Goggles/glasses. This is a device equipped with a suitable detection equipment which enables the computer to deduce the screen location at which a user is looking, then the user can action the selection of a particular object that he or she is looking at by winking or blinking (Seyer 1991, Barker 1993).

Voice input. Voice driven pointing devices are based upon a limited repertoire of spoken commands such as 'next', 'go', 'up', 'down', 'left', 'right' and so on (Seyer 1991, Barker 1993).

4.7.2 Hypertext and Artificial Intelligence

Combining artificial intelligence techniques with a hypertext resource is being used to extend and improve the types of facility that can be made available within non-linear information systems (Barker 1993). Some AI technologies, for instance rule-based expert systems, are already mature; some like blackboarding and neural networks, are beginning to find wider acceptance (Littleford 1991).

One of the most popular use is to combine the capabilities of an expert system with those of a hypertext facility. Expert systems and hypertext are complementary methodologies. As expert

systems explain their decisions to users, the rules on which the expert system is based often seem inadequate. Hypertext can be connected with rules so that textbook information can be offered to the user when the knowledge behind the rule is questioned. From the other side, hypertext systems can be extended by including expert system techniques within them that help guide the user to relevant information in the database (Rada 1989).

Some commercial products are appearing which combine aspects of AI and hypertext. They do so in different ways and sometimes for different reasons, but nonetheless the fusion of technologies is taking place.

Some of these products are: Nexpert Objects from Neuron Data, HyperTrans from Texas Instrument Inc., KnowledgePro Windows from Knowledge Garden Inc., Hyperbase from Cogent Software Ltd, and many more (Littleford 1991).

4.7.3 Hypertext and Telecommunications

As Barker (1993) state, development in hypertext are influenced by telecommunications facilities in three ways. Essentially, these relate to: 1) the global distribution of hypermedia materials; 2) the provision of shared remote access to resources; and 3) the creation of distributed hypermedia systems. Each of these is briefly discussed below.

Obviously, the availability of a telecommunications network means, in general, that information can easily be moved from one location to another. This implies that such networks can therefore be used to facilitate the dissemination of a hypermedia knowledge corpus to multiple users in a way that is independent of their geographical location. This also means that it now becomes very

easy to share hypermedia material, provided the information carrying capacity of the network (that is, its bandwidth) is sufficient to support the transfer of the embedded multimedia information within a reasonable time-scale.

Alternatively, if the actual dissemination of a hypermedia system is not required, a telecommunications facility can be used to provide distributed access. That is, users who are spread out in different (possibly remote) geographical locations can each simultaneously access (and possibly update) a centrally-located hypermedia knowledge corpus. Obviously, for sensitive information it may be necessary to impose various sorts of access control mechanism that will allow users only to gain access to information for which they have been granted access rights. It may also be necessary to impose charging mechanisms.

As well as facilitating dissemination and distributed access, a telecommunications network can also offer the possibility of implementing a distributed hypermedia system. Such a system is one in which different sections of the overall knowledge corpus are located at different geographical points.

CHAPTER FIVE

DEVELOPMENT OF A HYPERTEXT BASED COURSEWARE

5.1 Design Considerations

The development of a successful hypertext-based courseware depends critically upon the adoption of a sound and systematic design strategy that is based upon the use of well-founded guidelines and established design procedures (Barker 1993). These guidelines can be used to specify and interrelate the various steps and processes that are involved.

In this study we shall see different courseware and hypertext development models. The courseware development models are used to identify the various steps and processes in developing a courseware while the hypertext development models are used to identify the steps and processes in developing a hypertext application. A combination of steps and processes from the two applications are identified and used for the current work.

When designing computer-based instruction, designers are engaged in two activities: 1) creating a series of information screens and 2) organizing them into a certain order (Yang and Moore 1995). These two activities are well represented in the design of hypertext courseware which is composed of two parts, nodes and links.

Node design deals with the creation of discrete information screens while the link design deals with the connections between them. Usually, an information screen contains multiple stimuli. The significance of each stimulus is different. Therefore, it is the main consideration of node design to guide learner attention to the important content and thus help them achieve the best learning effect. The design of each screen is critical. However, only when these screens are well designed and presented in an appropriate order is the learners' cognitive construction facilitated (Yang and Moore 1995, Shneiderman, Kreitzberg, and Berk 1991)..

As Barker (1993) stated, before detailed design of a hypertext courseware can commence, it is important to resolve a number of 'top-level' design decisions relating to the overall courseware design; these decisions will significantly influence the way in which subsequent stages of the design will unfold. Some of the important decisions that have to be made will relate to answering the following basic questions:

- How will the product be accessed by the user?
- What will the final 'package' consist of?
- What type(s) of user is the product aimed at?
- What form of access control will be needed?

The characteristics of the delivery platform (such as pointing mechanism, storage medium, display device and so on) will dictate how the user accesses the product. This decision will also influence certain aspects of design mainly, the nature of the end-user interface that is produced (Barker 1993, Shneiderman, Kreitzberg, and Berk 1991).

Because the final hypertext package will usually consist of more than just interactive computer software, it is important to decide what other supporting materials will need to be supplied and what format they should take. In many situations a user-guide will be needed. Sometimes one or more workbooks may also need to be provided (Barker 1993).

An important design decision that has to be made at the start of a hypertext project is whether the product is aimed at single or multiple types of user group (such as novices, experts, casual users and so on). This will influence whether or not a multi-level design strategy is needed and, if so, which 'user models' should be employed. At this stage some thought must also be given to access control with respect to who will be allowed to read information from the application and who will be allowed to write information to it (Barker 1993).

5.2 Node Design

A node is the smallest piece of a hyper document that can be addressed by a link (Berk and Devlin 1991). Littleford (1991) defines it as a small portion of the document which covers one concept.

Node designing of hypertext courseware are mainly based on psychology, aesthetics, and human-computer interface design. Node design can be classified into the following categories: language usage, highlighting techniques, operating direction, and screen layout (Yang and Moore 1995, Barker 1993).

5.2.1 Language Usage

Since most courseware is used without direct assistance from outside tutors or instructors, a carefully crafted language is critical to the effective communication. The following guidelines can be applied to text as well as other information formats (Jones and Steinberg 1990, Yang and Moore 1995).

Language should be used at the appropriate reading level. The reading level of a language concerns not only the ability of learners, but also their cultural backgrounds and previous experience with the subject content. Commercial courseware producers often neglect the importance of appropriate language. They sometimes appear to understand very little about the potential learners. Emphasis is sometimes placed only on content expertise and programming skills, not on the formal instruction (Alessi and Trolip 1985 quoted in Yang and Moore 1995).

To help learners understand the content knowledge, courseware should keep language clear and specific. Technical terms are confusing and difficult for new learners. Before they are used, clear explanations have to be provided. An abbreviation garbles communication, especially when its full explanation is not on the same screen. Since hypertext allows learners to jump from node (content point) to node dynamically, it is crucial that each node be independent and self-explanatory. Technical terms and abbreviations should be avoided as much as possible (Jones and Steinberg 1990, Yang and Moore 1995).

Spelling, grammar, and punctuation are important in communicating the content information. However, the layouts of information are also valuable for they arouse and maintain learners' attention.

Indentation or blank lines signifies the beginning of a new paragraph. Upper-and lower-case text is preferred. Margins and line spacing are important for readability. These text layouts, although not related to the content, facilitate the cognitive process (Seyer 1991, Yang and Moore 1995).

Redundant expressions are inappropriate since the computer screen has only limited space. To make courseware efficient, the language should be to the point. Say only what needs to be said; use short words and short sentences.

A lean language uses not only concise syntax, but also simple expressions. It allows only one theme in each paragraph or on one screen. A lengthy and complex paragraph will "overload memory, cause confusion, and decrease interest" (Yang and Moore 1995).

5.2.2 Highlighting Techniques

There are many ways to direct learner attention to the emphasized information on the screen. But without reflective consideration, they may annoy learners.

It could be misleading to highlight a great deal of information on a single screen. Interface experts have suggested limiting the use of highlighting techniques to no more than three fonts, four text sizes, or four colors on a single screen (Jones and Steinberg 1990, Yang and Moore 1995).

There are conventions which are widely used to highlight information in traditional media. As a relative new medium, hypertext courseware should conform with the existing conventions. Violating these rules may cause confusion and decrease reading speed (Yang and Moore 1995).

Highlighting formats can provide extra cues to learners if they are used systematically. Those cues are particularly useful in conveying the structural information of the content. For example, a courseware unit uses italicized text for key terms. When learners are looking for terminology, they do not have to read through the whole text. They just skim screens by looking for the italicized text (Jones and Steinberg 1990).

5.2.3 Operating Directions

User-friendly operation can ease the anxiety and fear of learners. They also save learning time and effort.

Operating directions should be clear and specific. Ambiguity causes uncertainty and misunderstanding. For example, even if a system allows learners to proceed by hitting any key, it is better to tell learners to hit a specific key. Specific direction minimizes the mental effort and avoids possible troubles (Jones and Steinberg 1990, Yang and Moore 1995).

Operation should be consistent. Consistency of operation means universal position and universal meaning. The program operating directions should be placed in the same area and the same operation should initiate the same function. If the directions are located in different places on screens, learners must spend time to look for them. A program operation is frustrating if it means

different things in various contexts (Jones and Steinberg 1990, Barker 1993, Yang and Moore 1995)..

The design of program operation must meet the various abilities of learners ranging from novices to experts. In addition, during the navigation in courseware, learners should be allowed to access the operation directions at any time. An on-demand assistance is an important characteristic of a user-friendly system. It eases the memory load of learners. It is also adaptive to both novice and expert learners (Jones and Steinberg 1990, Barker 1993, Yang and Moore 1995).

5.2.4 Screen Layouts

To motivate learners, courseware screens should be artistically pleasant. Nevertheless, an effective screen layout should not attract learner attention to decorations or unimportant information. The screen design should be designed to decrease the mental burden and achieve better learning effects.

Since various types of messages co-exist on a screen, it is desirable to divide the screen into several functional areas. Each functional area uses fixed layout and location on the screen. A possible screen layout list includes: title, text, graphic, operation, or dialogue window. These functional areas make the appearance of screens neat and clear. The fixed locations of these areas also minimize unnecessary eye movements (Barker 1993, Yang and Moore 1995).

If a series of screens share one layout, the standardized format provides extra structural information and facilitates the learning process. In these screens, common objects remain static.

Therefore, the learner only pays attention to the changed objects instead of the whole screen (Yang and Moore 1995).

Cluttered screens increase mental burden and cause tension. Leaving appropriate space in margins, between lines, and between functional areas will have aesthetic and instructional value. Color also has a decided effect on learners' visual comfort (Barker 1993).

5.3 Link Design

A link is a pointer from one hypertext node to another. The presence of a link alerts a hypertext reader that there is a relationship between the node that is the source of the link and the link's target node.

The link design of hypertext courseware can be addressed from two perspectives, content organization and presentation flow. The content organization discusses the interrelationships on the subject matter aspect. The presentation flow addresses the interrelationships on the aspect of instructional process (Yang and Moore 1995).

5.3.1 Content Organization

An extended description of knowledge may overload memory and increase the difficulty of understanding the texts. The comprehensive knowledge should be broken down (chunking) into

many manageable fragments (Barker 1993, Yang and Moore 1995). It can be divided by the presentation formats, such as video, graphics, sound, or text. It can also be divided by the informational functions such as examples, principles, applications, or questions (Seyer 1990).

There are many ways to conduct the content breakdown. Merrill (1987), quoted in Yang and Moore (1995), identifies four primary presentation forms (PPFs) for computer courseware: expository instance (example), expository generality (rule), inquisitory instance (practice), and inquisitory generality (recall).

When constructing instructional fragments, a designer should consider a Performance-Content matrix to classify the instructional outcomes. Then, according to the performance level of the outcomes, the designer looks at a Performance-PPF Consistency table to find the appropriate primary presentation forms for presentation, practice, and performance. Again, according to the content category of the outcomes, the designer looks at a Content-PPF Consistency table to find the prescription for each primary presentation form (Yang and Moore 1995).

In addition to these primary presentation forms, Merrill (1987) also suggests secondary presentation forms. These forms do not come directly from the analysis of instructional content, but they direct learners to process the content information and thus the learning results. These terms include: enhancement, prerequisite information, alternate representations, mnemonics, correct answer feedback, feedback, and others.

Another way to chunk information into smaller pieces as suggested by Shneiderman, Kreitzberg, and Berk (1991) relates to identifying subsidiary topics and details. They suggest the following techniques.

- Instead of discussing a subsidiary topic which is not the main subject of a node, just allude to it and designate a phrase that refers to it as a link. Make that topic the subject of its own or at least give the topic a description which can be called up by learners. If learners are interested in more information about this peripheral topic, they can follow the link.
- Rather than including detailed information in a node, we can simply reference it and create a separate node for it. This shields learners from unnecessary details, but provide a path they can follow when it seems relevant.

Shneiderman, Kreitzberg, and Berk (1991) stated that the organizing principle is to present just enough information at each stage and provide details on demand. They provide the following algorithm for the design of hypertext links.

- Identify the domain of discourse so that a context is established. Meaningfulness is highly dependent upon context. For example, the phrase "I want to press a suit" has a very different meaning depending upon whether we are talking to a lawyer or a tailor.
- Identify pre-existing knowledge which learners need for complete understanding. Make such knowledge available through links so that the learner can access it and process the remainder of the information meaningfully.
- Present the information in logical sequence so that the information that comes first provides context for the information that follow.

- Identify details that the learner can obtain to elaborate the information in the initial presentation. When constructing a detail level, apply these same rules to it.

When designers have constructed all the information fragments, these fragments are put into a knowledge base for future retrieval. In hypertext systems, there are three basic organizations for storing and retrieving these informational fragments: unstructured, hierarchical, and network as shown in Figure 5.1.

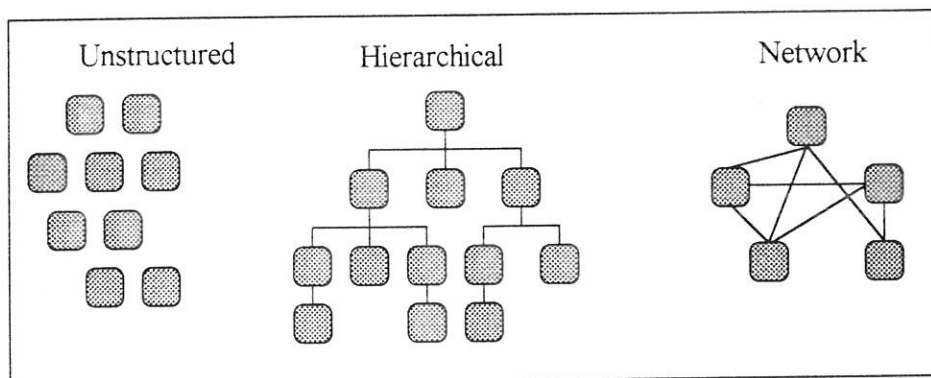


Figure 5.1 Three basic organizations of hypertext courseware

Unstructured knowledge bases store all information fragments randomly. Learners have to type a key to start the link between screens. They give learners the maximum flexibility of association at the price of being fully responsible for the success of their own learning. Electronic encyclopedias and dictionaries are examples. Hierarchical knowledge bases store information fragments in a fixed, tree-like sequence. Strict hierarchical systems allow learners to go only one-step upward or downward in the navigation. They are menu-driven information systems. In network knowledge bases, a node links to others based on the relationships between them. The complexity of the network structure depends on how many interrelationships between nodes exist.

For hypertext systems used in computer industry and commerce, the most popular organizations are network or unstructured formats. For traditional computer-based instruction, the content organization is linear or hierarchical. Since complex structures easily cause the loss of learners in hyperspace, a mix between hierarchical and network structure will be appropriate for most subject contents. The complexity of content organization is based on the nature of subject matter, learner characteristics, and learning objectives (Yang and Moore 1995).

5.3.2 Presentation Flow

Basically, learner control is a fundamental feature of hypertext courseware. Learners dynamically decide where to go, what to see, when to stop, and how to organize information. However, many learners are sidetracked to view unimportant information and are lost in navigation (Yang and Moore 1995, Barker 1993, Littleford 1991, Gay and Maruz 1991). Therefore, to achieve better results, certain presentation flows are proposed to limit the learner control of navigating in the hyperspace.

The presentation flow of courseware can be viewed from two aspect. A broader view of the presentation flow concerns the whole instructional process, while a narrower view concerns only the presentational sequence of the specific content (Yang and Moore 1995).

Gagne and his associates have devised a series of instructional events to correspond to the needs of learning process (Gagne, Briggs, and Wager 1992). According to this theory, no matter what the subject matter and learning objectives are, the critical instructional elements remain the same.

Table 5.1: Internal Learning Needs with External Instructional Events

Internal Learning Process	External Instructional Events
Reception	Gaining attention
Registration	Informing learner of the objective
Selective perception	Stimulating recall of prerequisite learning
Rehearsal	Presenting the stimulus material
Semantic encoding	Providing learning guidance
Retrieval	Eliciting the performance
Response generation	Providing feedback about performance correctness
Performance	Assessing the performance
Control	Enhancing retention and transfer

To enhance hypertext courseware, as many instructional elements as possible should be included in its presentation flow.

A presentation flow proposed by Yang and Moore (1995) for a hypertext courseware is shown in Figure 5.2. This presentation flow begins with the title page. The title page should be artistically designed to arouse learner interest. The indication of course name in the title page will also arouse the mind set for learning the course content. The next page presents learning objectives. The objectives should clearly communicate the expected learning result of the courseware. It is desirable to allow learners to access this page at any time they need.

After knowing what is expected, learners can select topics from the content menu. The tutorial screens of each topic should present clear explanation, refer to previous experience, and provide suitable guidance. After tutorial screens, each topic should have a practice section and provide informational feedback. Tutorial material, practice, and feedback should be provided for each topic.

When learners feel they have finished the tutorials and practice opportunities, they then can take a performance test. This test should evaluate their achievement and provide remediation. Once the performance has reached the criteria indicated in the objectives, learners can proceed to a summary and conclusion. In the end page, statistical information about the learning process and suggestions for further studies are presented. It provides reflection and recommendation concerning the learners' cognitive strategies and future transfer.

Generally, the presentation flow is somewhat fixed for a self-sufficient instructional medium. However, since hypertext courseware is highly interactive, the design of presentation flow should allow flexibility in the sequence of tutorial, practice, and performance test.

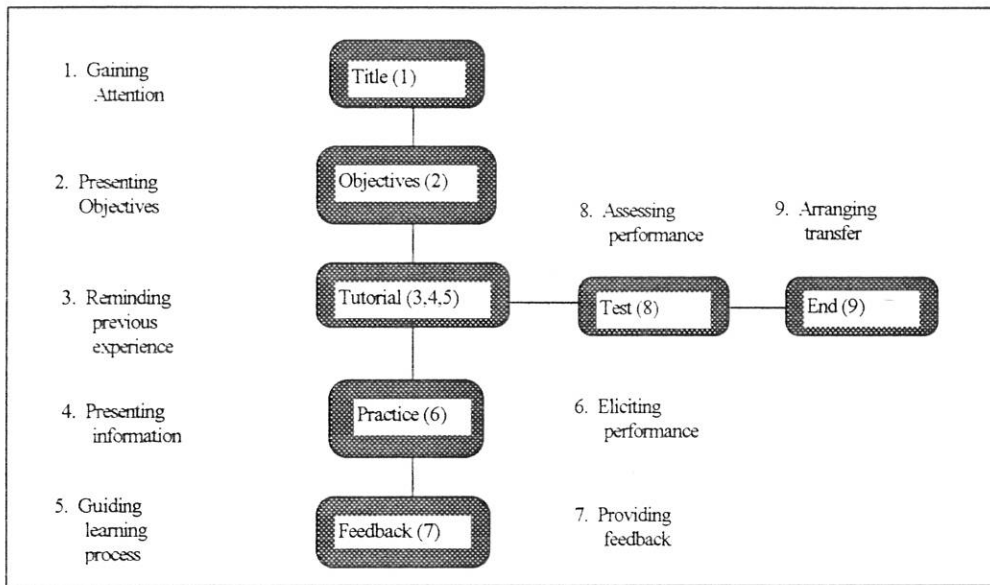


Figure 5.2 Presentation flow with instructional events

In this study this presentation flow will be adopted. In the following section we will see some courseware development and hypertext development models to have a general view of the steps and processes involved.

5.4 Development Models

A method suggested by Pistorius et al. (1992) consists of five phases.

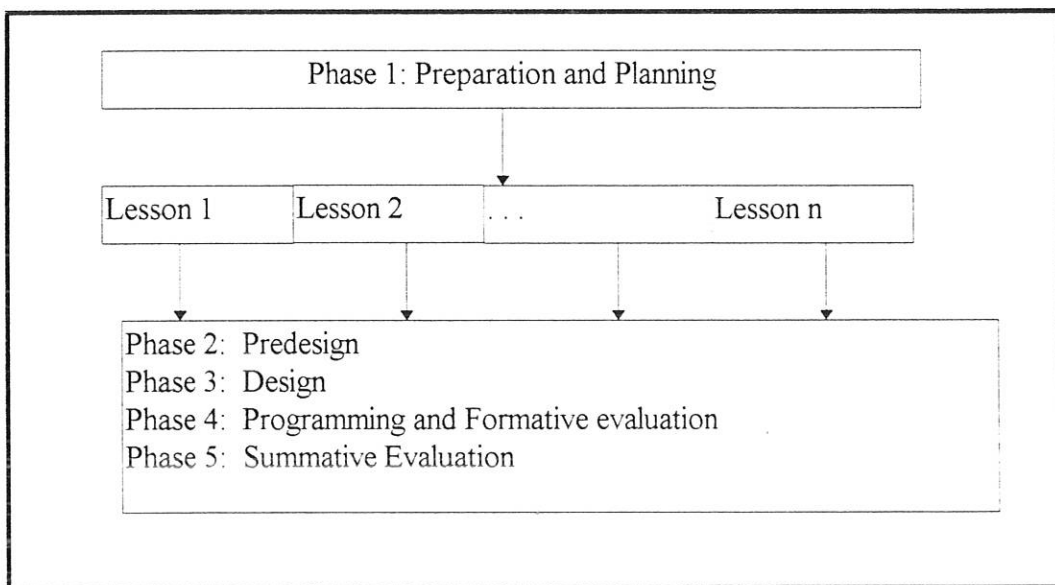


Figure 5.3 Courseware design model

Each of the five phases is made of the following detailed activities.

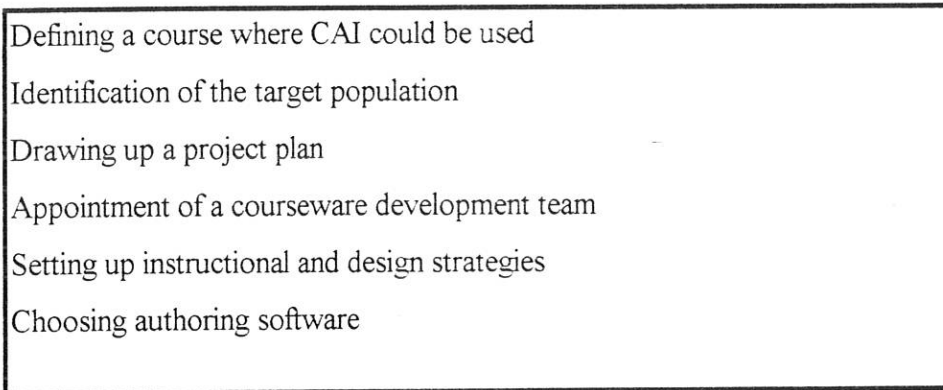


Figure 5.4 Phase 1: Preparation and Planning

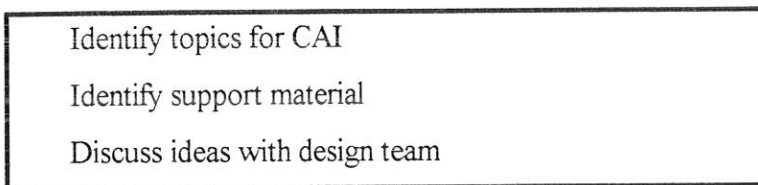


Figure 5.5 Phase 2: Pre-design

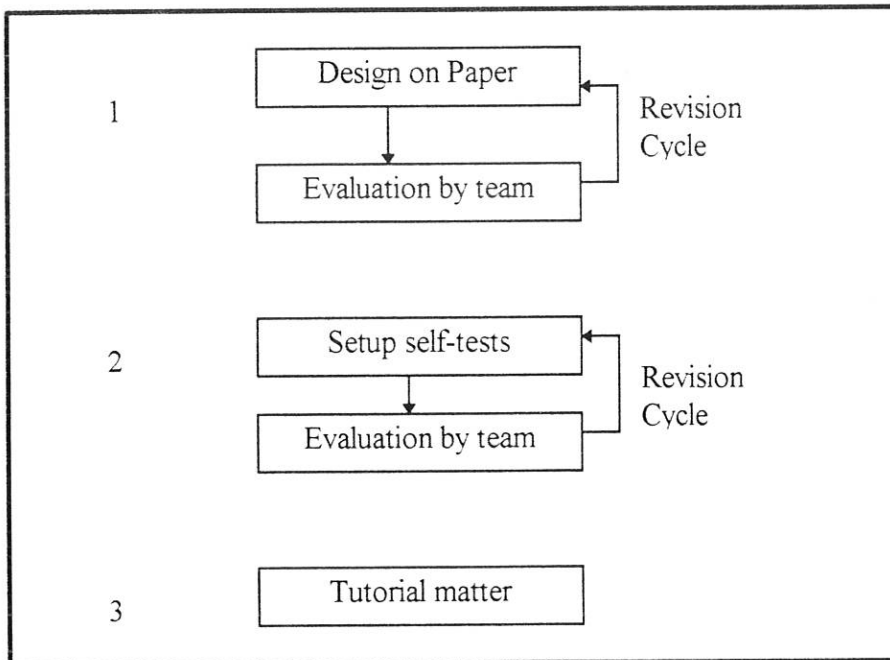


Figure 5.6 Phase 3: Design

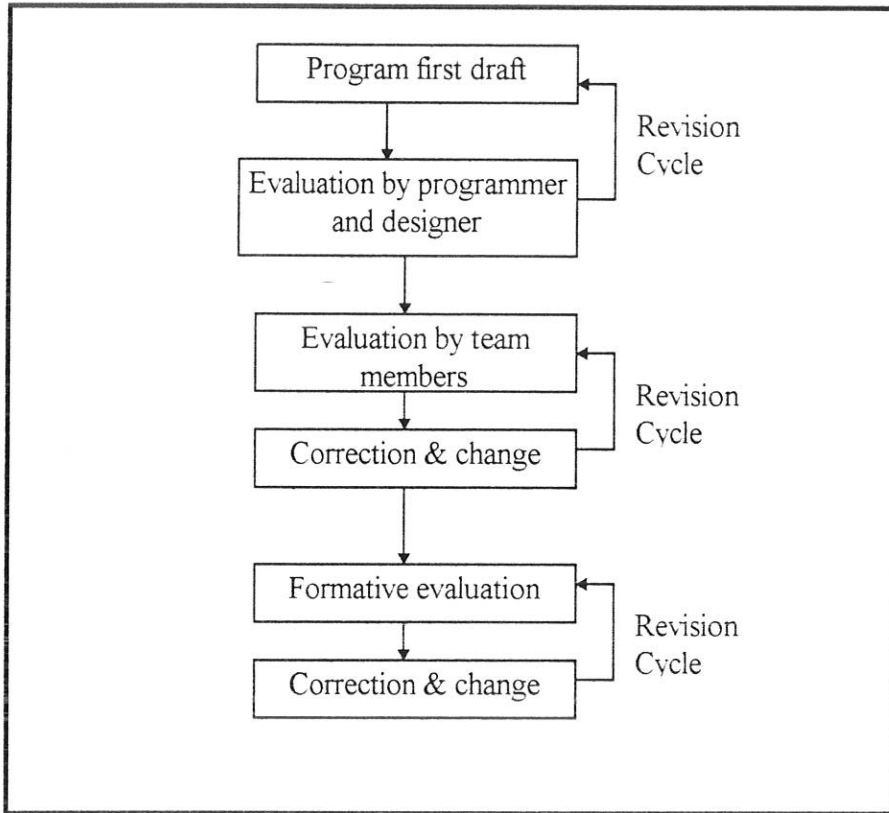


Figure 5.7 Phase 4: Programming and formative evaluation

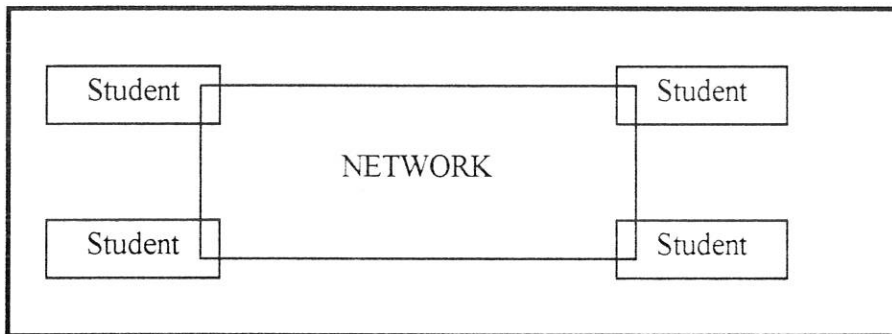


Figure 5.8 Phase 5: Summative evaluation by means of CMI

Another model advocated by Barker and Manji (1992) dictates that, a typical courseware engineering exercise include: *need analysis; requirement specification; instructional design; prototyping courseware; implementation; evaluation and testing; bulk production; distribution and marketing.*

Peter Chandra (Tagg and Lovis 1988,) suggests the design team approach to the development of computer based learning (CBL) courseware design. This model (Figure 5.9) depicts the

different expertise provided by team members for the various design stages. The design team approach relies heavily on the effective communication between different members of the team including up-to-date paper work and documentation.

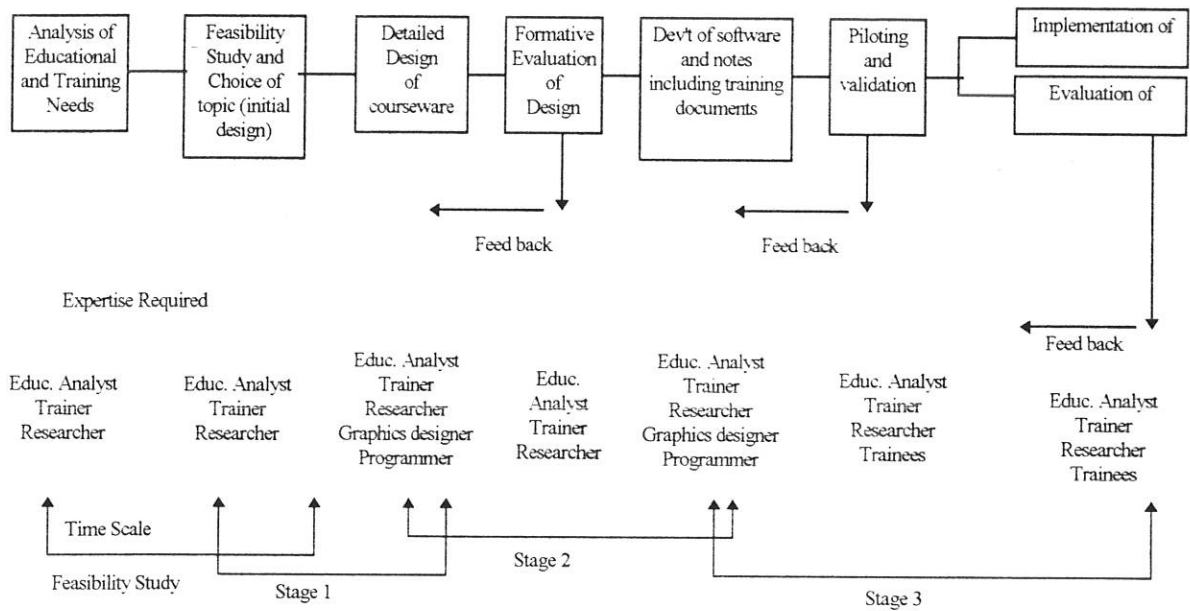


Figure 5.9 The team approach to CBL courseware design

Another more sophisticated model suggested by Ahmed-Oumer (1993) is the Computer Assisted Courseware Engineering (CACE) for industrial application called AGEDI (Atelier de GENie DIDacticiel).

This Model takes advantage of advances in information technology, on computer based learning and instruction. The main components of AGEDI are the Intelligent Computer Assisted (ICAI) module (application Layer), the hypermedia system (presentation layer) and the interaction supervisor (control) which chooses the best interaction mode in a learning session: free, guided or mixed one.

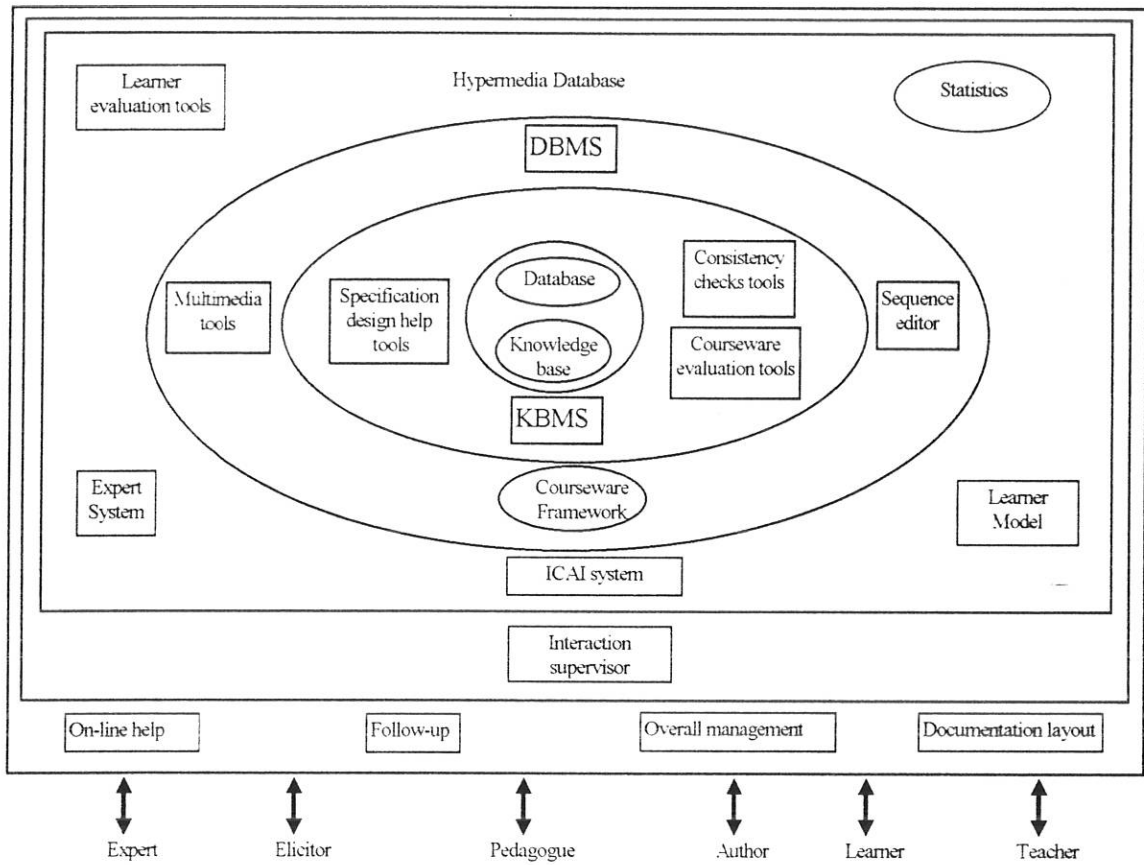


Figure 5.10 Overall architecture of the computer-assisted engineering AGEDI (Ahmed-Oumer 1993).

Another model in the field is the Instructional System Design (ISD). The general term that refers to the systematic approach to the design and development of instructional materials regardless of the medium is Instructional Systems Design (O'Neil 1981 and Rosenberg, 1982). It provides a procedure for systematically identifying and manipulating significant components which make up the instructional process, the goals of which are increased learning and improved performance .

Although there are many variations on ISD methodology, most are based on five phase model, i.e., analysis, design, development, implementation, and evaluation as indicated in the figure 5.11.

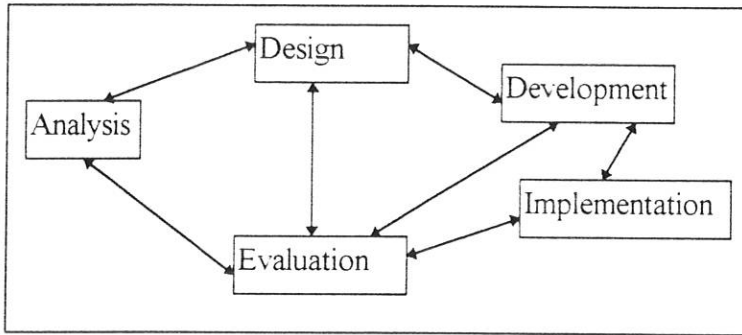


Figure 5.11 Major phases of ISD

The details of the input to and the output from each phase in the ISD model is given in figures 5.12 through 5.15. The evaluation measures are also indicated.

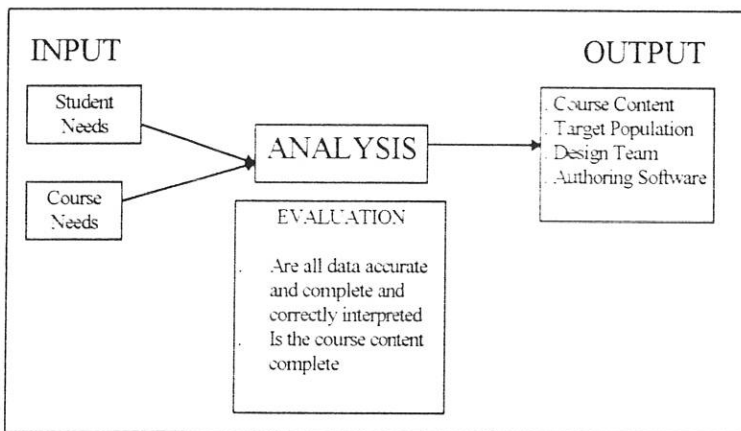


Figure 5.12 Analysis phase

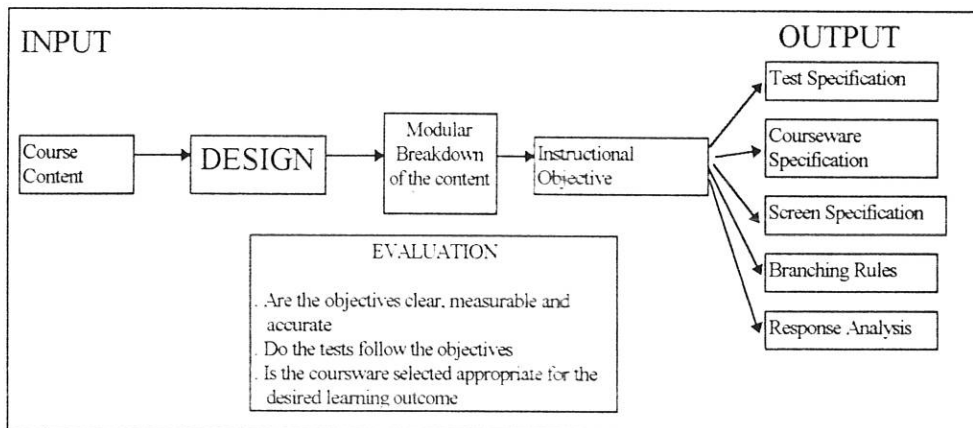


Figure 5.13 Design phase

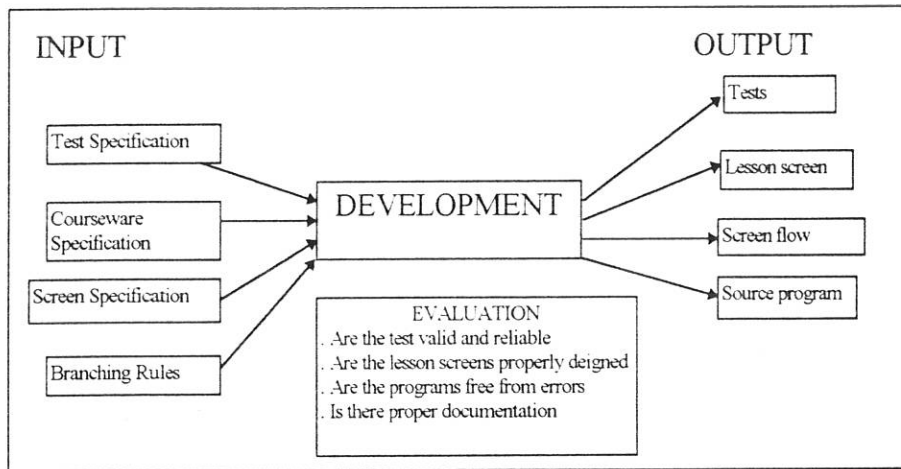


Figure 5.14 Development phase

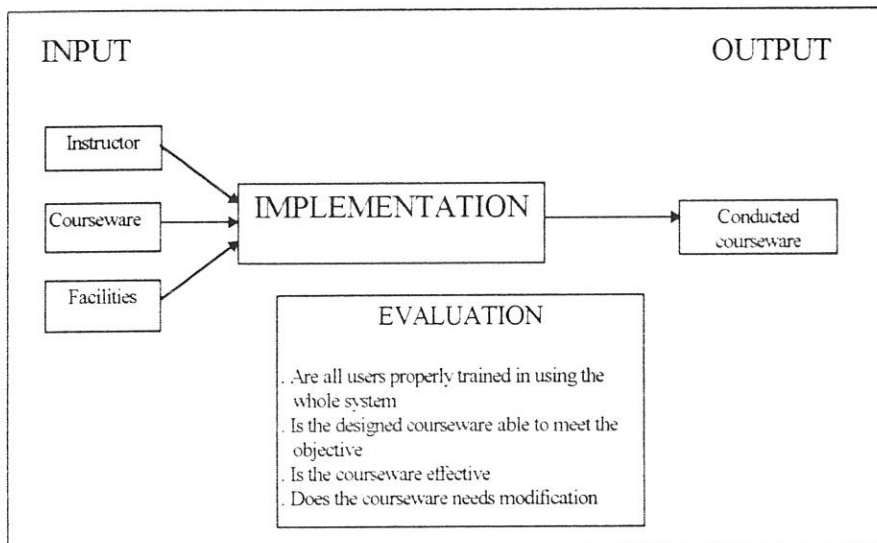


Figure 5.15 Implementation phase

Some of the weakness as reported in the literature about the different models discussed above include:

1. The approach suggested by Peter Chandra is reported to has been hampered by the lack of authors who have the technical appreciation of how their ideas and teaching strategies can be implemented on a computer (Tagg and Lovis 1988).

2. AGEDI practicality is reported to have been limited due to the unavailability of the sophisticated technological facilities/resources required by the model in many of courseware development projects (Ahmed-Oumer 1993)
3. The ISD approach is criticized for specifying only what is to be done and not how to do it (O'Neil 1981).

Since the intended courseware is a hypertext based we shall see a model for developing hypertext applications. A basic hypertext development model given by Barker (1993) is shown in figure 5.16.

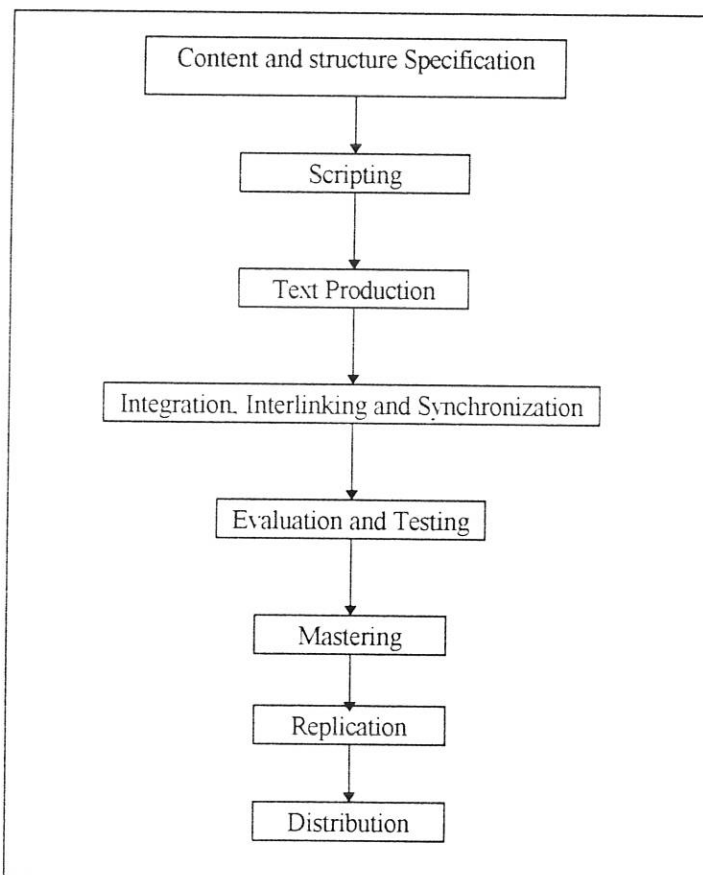


Figure 5.16 Basic hypertext development model.

One fairly obvious limitation of this model is its linear nature. While many well-defined hypertext projects are likely to follow a linear development strategy, many others will not. Indeed, many more complex development projects will often involve the creation of a 'prototype' system (Barker 1993). Inherent in this approach will be the use of iteration and back-tracking in order to identify the most beneficial way to proceed.

Despite its obvious limitations, Figure 5.16 serves the useful purpose of indicating two important aspects of a hypertext development project: first, the nature of the basic processes involved and second, the rough order in which these different processes should be performed. Many of the basic processes listed in the figure, such as content and structure specification, scripting and so on, will be similar to those involved in the production of interactive training materials for delivery by computer (Bergman and Moore 1990, Dean and Whitlock 1992, and Harrison 1991).

The above approaches deal with problems in a sequential and linear fashion which real life software projects rarely follow. Another problem of these approaches is that the software designer is actually introduced to the product only after the implementation so that change request are likely to build up from that point. It is possible that a significant gap exists between the designer's expectation and the product capabilities (Aktas 1987).

5.4.1 Using prototypes

Because hypertext systems can be far more complex than conventional interactive systems, the linear development model described earlier and shown in Figure 5.16 is frequently an

inappropriate one to use. Instead, a cyclic, iterative development approach is often a more suitable one to adopt. Inherent in the use of this approach is the creation of a development 'prototype' (Barker 1993, Wong 1993).

Essentially, a prototype system is one which is constructed in order to explore and investigate the 'look and feel' of a product without having to undertake a complete system implementation. A prototype is therefore a partial implementation of a product which can be used to evaluate and test the basic system design and the appearance of a product before it is finally implemented. Prototyping provides valuable feedback information about the underlying design of a system and, if necessary, this information can be used to modify the design prior to completing the final implementation (Barker 1993).

Prototyping is a process that enables the developer to create a model of the software to be built. To prototype is to develop an executable version of a product which incorporates key elements of the final version but which is incomplete in many respects, for example, in terms of functionality, robustness, or exception handling (Wong 1993). With a prototype, software designers and developers can actually see what is possible and how their requirements translate into hardware and software. Quick prototyping is the process of quickly building and evaluating a series of prototypes and requires the availability of tools that offer modularity and plasticity. It allows one to create and test input designs, terminal dialogues, and simple procedures.

Typically, a design team that is developing a hypertext product would use a prototype system in order to test out their screen layouts, color combinations, button designs, link strategies, interaction methodologies and various other design parameters on just a part of the system. Then, if there are any obvious changes that need to be made these can be

undertaken before the final stages of implementation get underway. The use of a prototype can therefore be used to circumvent mistakes that may otherwise be costly to correct.

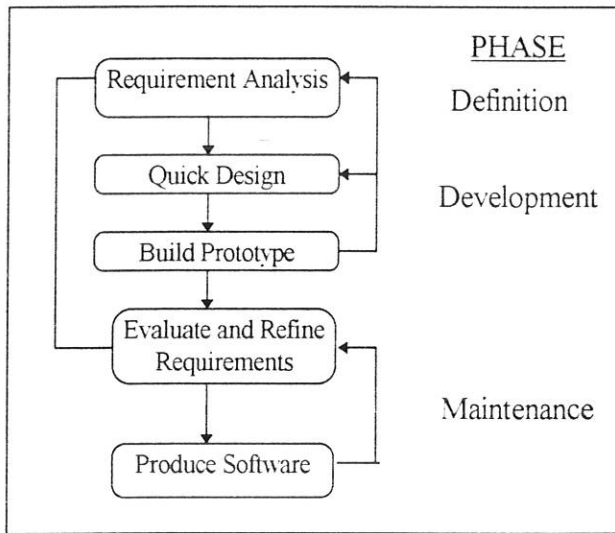


Figure 5.17 Prototyping

Like all approaches to software development, quick prototyping consists of three generic phases: definition, development, and maintenance (Wong 1993). The definition phase focuses on what. That is during the definition, attempts are made to identify what information is to be processed, what function and performance are desired, what interfaces are to be established, what design constraints exist, and what evaluation criteria are required to define a successful system. Thus, the key requirements of the system and the software are identified.

The development phase focuses on how. The development phase of the prototyping approach involves a quick design. The quick design focuses on a representation of those aspects of the software requirements outlined. Quick design leads to construction of a prototype.

The maintenance phase focuses on change that is associated with error correction, adaptation required as the software' environment evolves, and modification due to enhancements brought about by changing requirements. The maintenance phase

reapplies the steps of the definition and development phases, but in the context of existing software. Thus, the prototype is evaluated and used to refine requirements for the software to be developed. A process of iteration occurs as the prototype is tuned to satisfy the requirements until a software which meets the objectives has evolved.

In this study the model suggested by Yang and Moore (1995) is used for the presentation flow and content organization. For the development of the prototype software the quick prototyping paradigm is used.

CHAPTER SIX

PROTOTYPE DEVELOPMENT AND IMPLEMENTATION

This chapter describes the stages of development of the prototype courseware. It may be noted that the courseware has been developed by using the quick prototyping development approach. Measures to be taken for implementation of the courseware are also discussed.

6.1 The Definition Phase: Requirement Analysis

In this phase of software development the attempt is made to identify the information to be processed, the interfaces needed, the functions and performances desired, and the validation criteria required. This helps to identify the key requirements of the system and the software.

Hence, in reference to the current study, the course content for the courseware; the student requirement, to identify the entry level for the courseware which helps us to decide on the type of interface; the type of programming language used which can support the desired functions and performances of the courseware; and the team involved in the development to validate the content and the program are the matters of concern which had to be identified.

Since the course "Introduction to computers" is the first out of the various courses on the subject offered by NCIC, the trainees are not expected to have any prior knowledge of computer systems.

"Introduction to Computers", the course content for the courseware, is being offered at NCIC. The Amaharic version of the course is adopted from the book by Ato Negusu Atnafu (1993), who was a staff member at NCIC and now is a manager of CAROS Computer Service, with his consent. The content of the book and the consistency of the terminologies used in the book with those used at NCIC Amaharic software projects is checked by the Amaharic software development team of NCIC, which includes the researcher. Some topics in the course "Introduction to Computers" are not covered in the book. These topics were translated by the researcher and Ato Getachew Mengistu, an expert at NCIC, and added to the courseware.

The course is divided into five main topics and sixteen modules. The detailed breakdown of the topics is given in Chapter Two.

Courseware development is usually a team work which requires the expertise of different personnel. Some of the basic expertise needed are subject expert, instructional systems expert and programming expert.

For the purpose of this study and in due consideration of the constraints within which the study operates, a role playing team was formed. In addition to utilizing the researcher's teaching experience of the course and familiarity with programming, senior computer experts at NCIC were consulted in revising and modifying the lessons, and in programming the courseware. In addition, advisors of this thesis were involved 'instructional systems experts' in shaping the flow of the courseware.

The courseware, being hypertext based, required authoring languages capable of handling hypertexts. One of such software tools is KnowledgePro for windows which is available at SISA. In addition to being the only software available, KnowledgePro for windows is found to be particularly advantageous for the current work. Some of these advantages include its capability to be used from either a procedural or an object oriented approach. This is found important since the researcher is mainly familiar with procedural languages. Knowledge Pro comes with sample applications and sections of codes which illustrate how to perform certain common tasks. It also enables to paste the syntax of all of its functions to the application being developed. This greatly saved the development time needed for the courseware. It is capable of handling fonts supported by Windows which solved the problem of the incorporating of the Amaharic scripts.

However, the KnowledgePro editor doesn't allow to type fonts directly. This made it difficult to directly format text which doesn't use the system font. KnowledgePro provides a solution to this problem. The solution is to format the text with Microsoft Windows' Write and then convert the file into an ASCII file that can be read into the KnowledgePro editor.

Still when the text is displayed in the KnowledgePro editor it is displayed in the system font. However, when the application is run and the font is set to the font used in Write, the text is properly displayed.

6.2 The Development Phase

6.2.1 Quick Design

The intended hypertext-based courseware delivery platform is Microsoft Windows environment which is inherently capable of handling interaction by almost any brand of mouse. Therefore, the courseware can be accessed by the learners using a mouse. The decision to access the courseware with a mouse is also influenced by the nature of the Amaharic environment. The Amaharic environment is relatively new and there are different keyboard layouts (the keyboard layouts are not standardized). Furthermore, there are many shifting keys to be pressed which makes a keyboard interaction by novice users somewhat awkward.

As indicated in the analysis, most students are first time computer users. However, to make the courseware more interactive, to make revision of previously studied topics more convenient for second and third time users, and to make the courseware useful for anyone who needs to refresh his knowledge, the product was aimed at both novice and expert users.

The Amaharic language used in each node was made to be the one used in the student's day-to-day communication. Technical terms and abbreviations are explained in the same screen.

Since nodes can be accessed dynamically from different nodes, each node was made self-explanatory and independent. But to maintain the relationships of nodes, a link to a related information was given at the end of each of the related nodes.

Spelling and punctuation marks are important to communicate the content information clearly. Since there are no spelling checking facilities for Amaharic text, the spelling errors were avoided manually by going through the content repeatedly. Layout features such as margins, line spacing, blank lines for paragraph separators were used to maintain learner's attention.

Red colour was used to alert learners to do something more before branching into another lesson and to warn them that they should repeat a module because of failure in exercises.

The operating directions were designed in such a way that they give clear and specific instruction. They were given in Amaharic words or phrases to avoid ambiguity from the use of icons by novice users. The directions to go to the index, backward one screen at a time, to do exercises and performance test, and to access on demand help, were always located at the top of the screen on all windows.

The text colour for the lesson was designed to be black text on white background. The reactive or hypertext words or phrases were underlined. This enabled the text to be displayed in a wide variety of display screens. In view of colour monitors, alerting messages and hypertexts were given in red and green colors respectively. Furthermore, they were also given additional styles of italics and underline by considering monochrome screens.

Another important design consideration in a hypertext courseware is the link design. As described in the previous chapter, it addresses the interrelationships on the subject matter and the interrelationships on the aspect of the instructional process.

The instructional outcomes as categorized by Gagne (1977) are presented below:

1. Information
2. Intellectual capability
 - 2.1 Discrimination
 - 2.2 Concrete concept
 - 2.3 Defined concept
 - 2.4 Rules
 - 2.5 Problem solving
3. Cognitive strategy
4. Motor skill
5. Attitude

On the basis of these outcomes, one could have the following performance levels.

Recall a name

Explain how to do a task

Define a concept

Solve a theoretical or practical problem

Identify things

Perform simple physical acts

Perform skilled physical acts

Determine quality of a product or evaluate performance.

The subject introduction to computers is broken-down into five main topics. Each topic is further broken down into modules. Each module is broken down, according to the expected learning outcomes, to present a concept, information, and simple rule. Examples, rules, practical exercise, etc. were made separate nodes and were linked to main concepts.

Enhancements and prerequisite information for a given topic were also linked to a topic. Different synonyms were linked in such a way that direct learners to the main term. Incorrect answer feedback was linked to the appropriate topic to show students the correct answer.

Each screen of information or node was designed to present just enough information and details are linked to be accessed on demand.

As indicated before in this section the courseware is intended for both novice and expert users. For the novice users the courseware was presented in a guided manner starting from the first module to the last. But not to lose the spirit of hypertext, related topics within the same module were linked and thus can be accessed from any of the topics; links were made for elaboration and examples within the same topic. If novice users are not guided through the tutorial, they may interrupt the courseware before the intended learning objective is met or they may repeat the same path unnecessarily.

On the other hand, expert or experienced users can access any part of the courseware through the index. They can also see the content to identify what topic is available in which module.

The branching rules used for the courseware are, therefore, mastery and user control. In the mastery model pre- and post-tests were used to determine if the student achieved the desired mastery level. If the student hasn't achieved the desired mastery level remedial instructions and/or additional practice is given until mastery is achieved. In the user control model students are responsible for choosing their own sequence of learning.

Taking the nature of the majority of the learners into consideration, the presentation flow of the courseware was designed such that it follows the flow suggested by Yang and Moore (1995) and presented in the previous chapter. This presentation flow was designed to include all the nine instructional events.

Additional design consideration include, objective determination. Objective determination enables to know where one is going and at the same time to recognize the place when one gets there. It also enables to measure whether a learner reaches the desired level of performance or not. The objectives for each module should be given at the beginning clearly and in measurable way based on the expected learning outcomes. The Amaharic equivalent of terms describing performance such as locate, select, describe, etc., are used to specify the objectives. These terms are useful in facilitating specification of criterion in the form of a test for measuring the achievement of the objectives.

At the end of each topic students are made to work on exercises to decide whether they are capable of going to the next topic or not. The exercises were designed by considering the characteristics of the students. They were made to be selected by a mouse to reduce the need for keyboard entry. Responses as typing long sentences usually delay the learner's progress and may not be mostly convenient for novice users.

6.2.2 Prototype Development

The structure of this courseware is illustrated in figure 6.1.

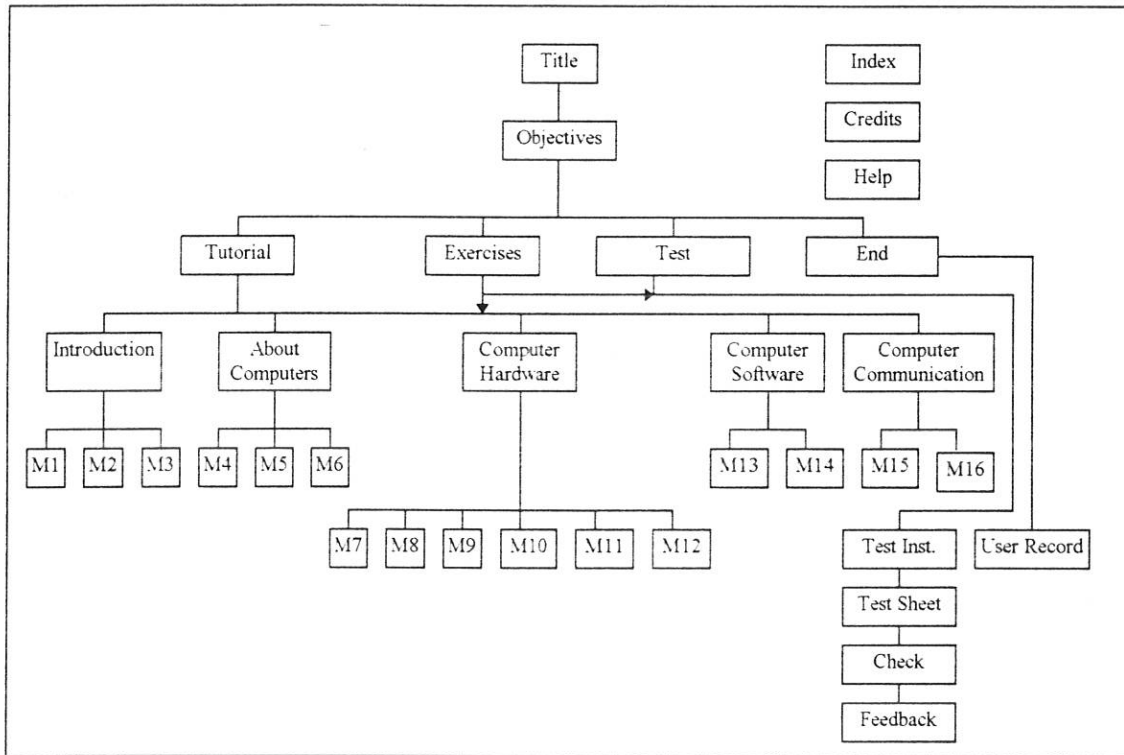


Figure: 6.1 The structure of the courseware

Figure 6.2 shows the development flowchart of the courseware which reveals the control. It is derived from figure 6.1 and follows the intended guidance for novice users. Expert users are provided with an index table which contains a list of hypertexts. The hypertexts in the index are links to key concepts in the courseware.

The screen flow of the courseware is given in figure 6.3. In this section, only the links between screens beginning from the title screen up to the end of the first lesson are given for demonstration purpose.

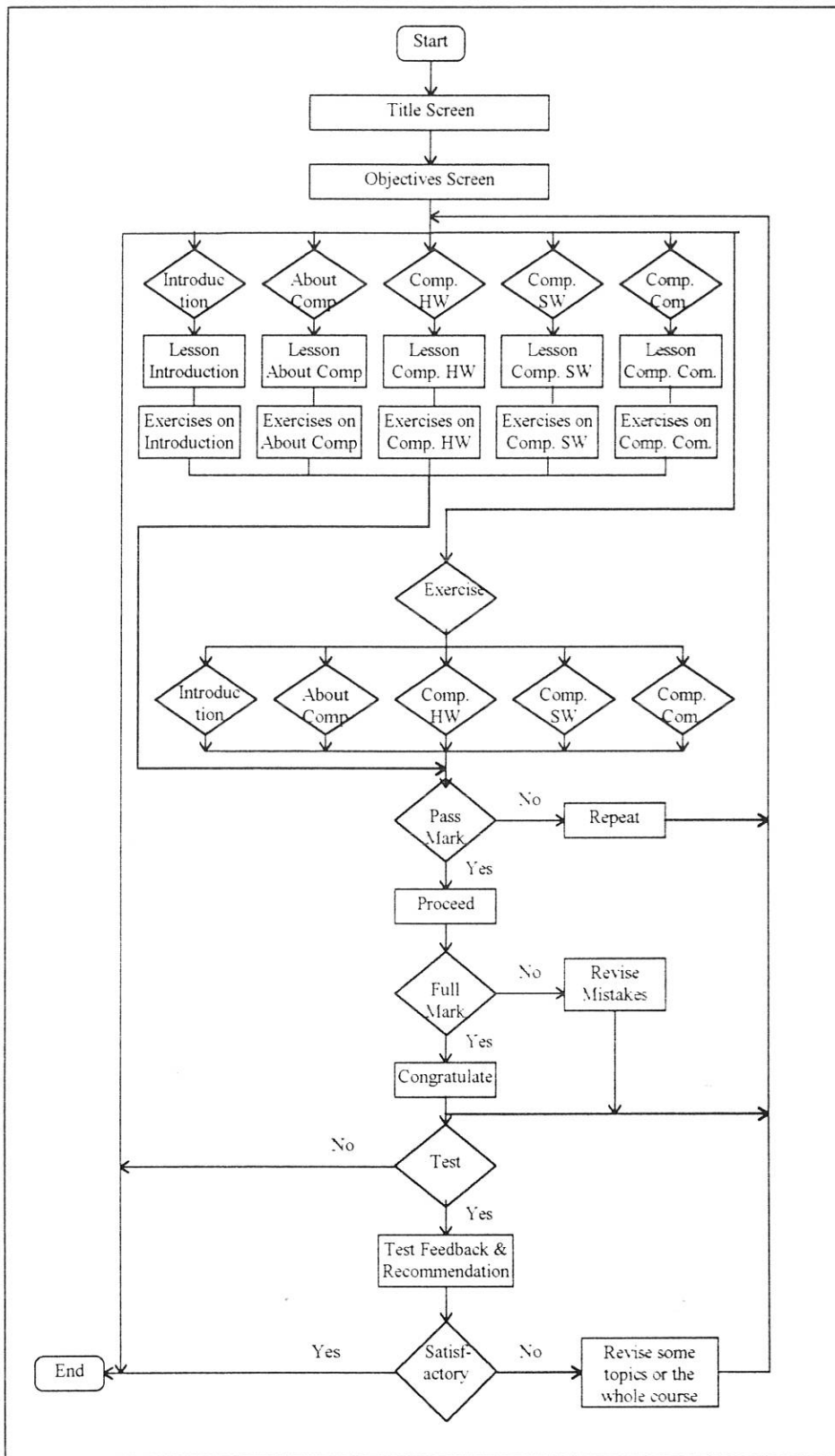


Figure 6.2 Development flowchart of the courseware

Detail explanation of the screen flows is presented below.

Screen TITLE

Function: Displays the courseware introduction

Linked to: Screen የክርስት ዓለማ

Screen የክርስት ዓለማ

Function: Displays objectives of the courseware

Linked to: Screen ማውጫ

Screen ማውጫ

Function: Displays the courseware content

Linked to: Screen መግቢያ

Screen ስለ ኮምፒዩተርኛ

Screen ኮምፒዩተር ሀርድዌር

Screen ኮምፒዩተር ሶፍትዌር

Screen ኮምፒዩተር ኮሙኒኬሽን

Screen ያርዕስት ማውጫ

Screen መግቢያ

Function: Displays the objectives of the topic መግቢያ

Linked to: Screen መረጃን ማቀናጀት overview

Screen መረጃን የማቀናጀት መመዘኛዎች overview

Screen መረጃን የማቀናጃ ልዩልዩ መንገዶች overview

Screen መረጃን ማቀናጀት overview

Function: Displays overview of the basic concept መረጃን ማቀናጀት

Linked to: Screen መረጃን ማቀናጀት elaboration

Screen መረጃን የማቀናጀት መመዘኛዎች overview

Screen መረጃን ማቀናጀት elaboration

Function: Displays elaboration on the concept መረጃን ማቀናጀት

Linked to: Screen መረጃን የማቀናጀት መመዘኛዎች overview

Screen መረጃን የማቀናጀት መመዘኛዎች overview

Function: Displays an overview of the concept መረጃን የማቀናጀት መመዘኛዎች

Linked to: Screen ትክክለኛነት

Screen ወቅታዊነት

Screen ፍቺያማነት

Screen መረጃን የማቀናጀት ልዩልዩ መንገዶች overview

Screen ትክክለኛነት

Function: Displays one of the three components of the concept መረጃን የማቀናጀት መመዘኛዎች

Linked to: Screen ወቅታዊነት

Screen ፍቺያማነት

Screen መረጃን የማቀናጀት ልዩልዩ መንገዶች overview

Screen ወቅታዊነት

Function: Displays one of the three components of the concept መረጃን የማቀናጀት መመዘኛዎች

Screen ተክለኛነት

Screen ፍቺያ ማነት

Screen መረጃን የማቀናጃ ልዩልዩ መንገዶች overview

Screen ፍቺያ ማነት

Function: Displays one of the three components of the concept መረጃን የማቀናጀት መመዘኛዎች

Screen ተክለኛነት

Screen ወቅታዊነት

Screen መረጃን የማቀናጃ ልዩልዩ መንገዶች overview

Screen መረጃን የማቀናጃ ልዩልዩ መንገዶች overview

Function: Displays overview of the concept መረጃን የማቀናጃ ልዩልዩ መንገዶች

Linked to: Screen መረጃን የማቀናጃ ልዩልዩ መንገዶች elaboration

Screen በሰው መረጃን ማቀናጀት

Screen በኤሌክትሮ ሜካኒካዊ መንገድ መረጃን ማቀናጀት

Screen በኤሌክትሮኒካዊ መንገድ መረጃን ማቀናጀት

Screen መልመጃዎች

Screen መረጃን የማቀናጃ ልዩልዩ መንገዶች elaboration

Function: Displays elaboration of the concept መረጃን የማቀናጃ ልዩልዩ መንገዶች

Linked to: Screen በሰው መረጃን ማቀናጀት

Screen በኤሌክትሮ ሜካኒካዊ መንገድ መረጃን ማቀናጀት

Screen በኤሌክትሮኒካዊ መንገድ መረጃን ማቀናጀት

Screen መልመጃዎች

Screen በሰው መረጃን ማቀናጀት

Function: Displays one of the three components of the concept መረጃን የማቀናጀት ልዩልዩ መንገዶች

Linked to: Screen በኤሌክትሮ ሜካኒካዊ መንገድ መረጃን ማቀናጀት
Screen በኤሌክትሮኒካዊ መንገድ መረጃን ማቀናጀት
Screen መልመጃዎች

Screen በኤሌክትሮ ሜካኒካዊ መንገድ መረጃን ማቀናጀት

Function: Displays one of the three components of the concept መረጃን የማቀናጀት ልዩልዩ መንገዶች

Linked to: Screen በሰው መረጃን ማቀናጀት
Screen በኤሌክትሮኒካዊ መንገድ መረጃን ማቀናጀት
Screen መልመጃዎች

Screen በኤሌክትሮኒካዊ መንገድ መረጃን ማቀናጀት

Function: Displays one of the three components of the concept መረጃን የማቀናጀት ልዩልዩ መንገዶች

Linked to: Screen በሰው መረጃን ማቀናጀት
Screen በኤሌክትሮ ሜካኒካዊ መንገድ መረጃን ማቀናጀት
Screen መልመጃዎች

Screen መልመጃዎች

Function: Displays the list of topics to select from for doing exercises on.

Linked to: Screen መመሪያ

Screen መመሪያ

Function: Displays instruction for doing the exercises.

Linked to: Screen በመግቢያ መልመጃ

Screen ስለ ኮምፒዩተር መልመጃ

Screen በኮምፒዩተር ሀርድዌር መልመጃ

Screen በኮምፒዩተር ሶፍትዌር መልመጃ

Screen በኮምፒዩተር ኮሙኒኬሽን መልመጃ

Screen በመግቢያ መልመጃ

Function: Displays exercises on the first lesson

Linked to: Screen ምዕራፉን ይደገሙ

Screen ወደሚቀጥለው ምዕራፍ ይለፉ

Screen ምዕራፉን ይደገሙ

Function: Displays Information to repeat this lesson

Linked to: Screen ማውጫ

Screen ወደሚቀጥለው ምዕራፍ ይለፉ

Function: Displays Information to proceed to the next lesson

Linked to: Screen ማበረታቻ

Screen የተሳሳቷቸው ጥያቄዎች

Screen ማበረታቻ

Function: Displays congratulatory complements

Linked to: Screen ማውጫ

Screen የተሳሳቷቸው ጥያቄዎች

Function: Displays list of mistakes which are linked to the appropriate topic.

Linked to: The appropriate topics

The prototype courseware was developed after a repetitive iteration which resulted in modifying of specification and coding. For example, the specification to format the Amaharic text using Microsoft Windows' Write was modified and Word 6.0 was used to format the text. It was found that the keyboard program of the Amaharic software Agafari for Windows, used in this study, could not function in Windows' Write, although it is functional on many windows based application programs such as Word 6.0, WordPerfect 6.1, Excel 5.0, Access 2.0, etc.

One alternative tried was to use a wordprocessor capable of saving files in a wide variety of text formats. Word 6.0 was found capable of handling both the Amaharic typing and the exporting of text into different text formats. Saving the Amaharic text in the format Text with Layout was found successful.

Although the Amaharic text is not readable in the KnowledgePro editor, some codes has been written in Amaharic with explanation given in English as a remark.

The courseware starts with a Title screen, which is artistically designed to create learner interest, by displaying pictures of a computer training situation, a disk, a printer, other related items, and the course name as in figure 6.4.

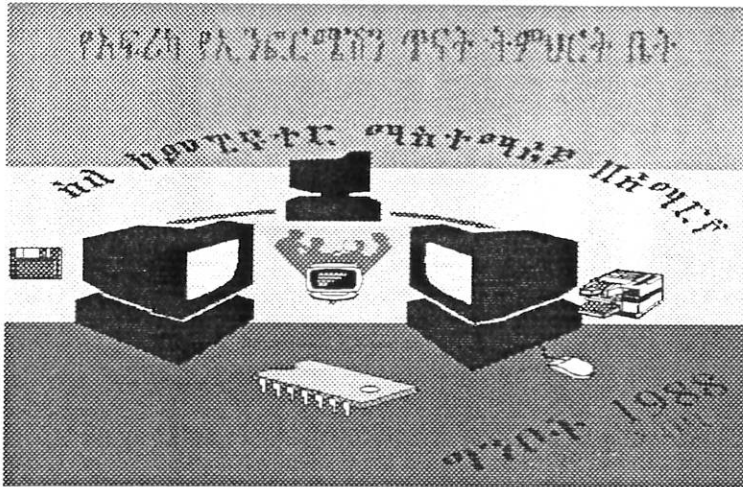


Figure 6.4 Title Screen

The Title page is followed by the Objectives screen which is designed to clearly communicate the expected results of this courseware (see figure 6.5). A button is dedicated to this screen to make it accessible from any window at any time. The objectives button is the first button in each screen.

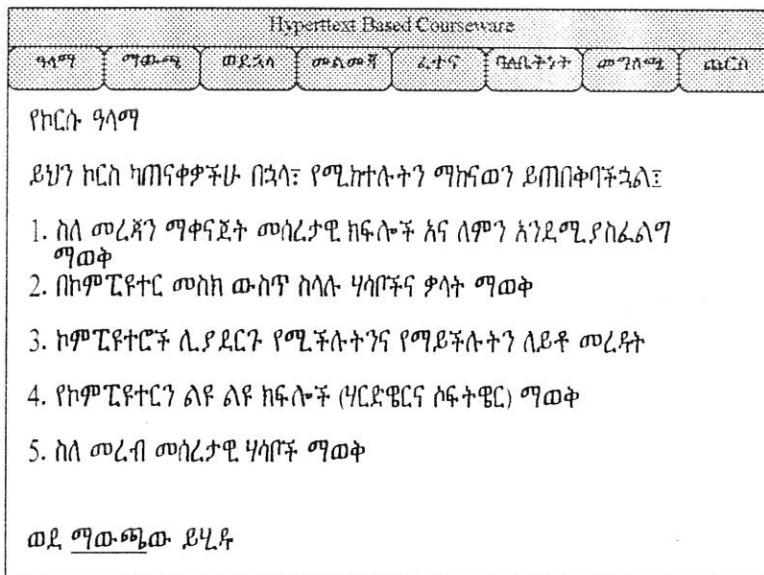


Figure 6.5 Objectives Screen

After letting the learners know what is expected, they are made to select topics from the contents screen. The contents screen provides the main topics of the course and the Index as shown in figure 6.6.

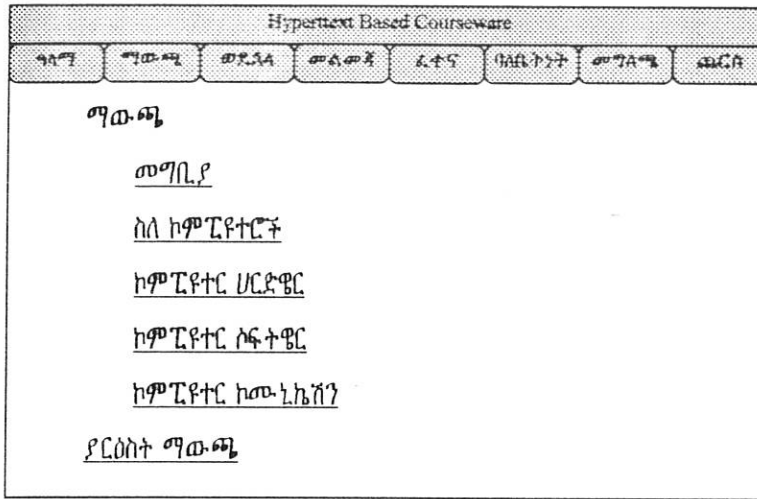


Figure 6.6 Contents Screen

When a topic is selected, the objectives screen for that topic with the main module names made hypertexts is displayed (see figure 6.7).

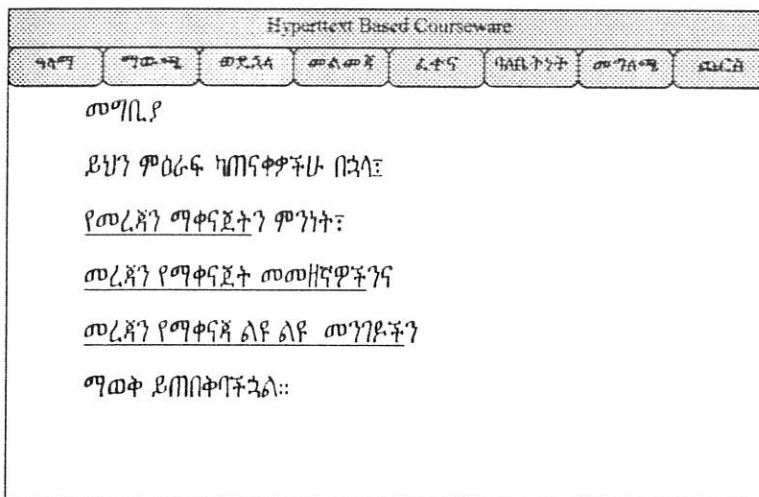


Figure 6.7 Main Topic Objectives Screen

From the main topic's objectives screen the students can start the learning process. In each module a basic concept is introduced. Elaboration and examples are made accessible. The tutorial screen presents clear explanation. They also provide suitable guidance by making links to related items. by alerting them to see related items before proceeding to a new topic, by providing a way to retrace their paths to refer to a previous topic. Figures 6.8, and 6.9 shows screens for overviews and elaborations.

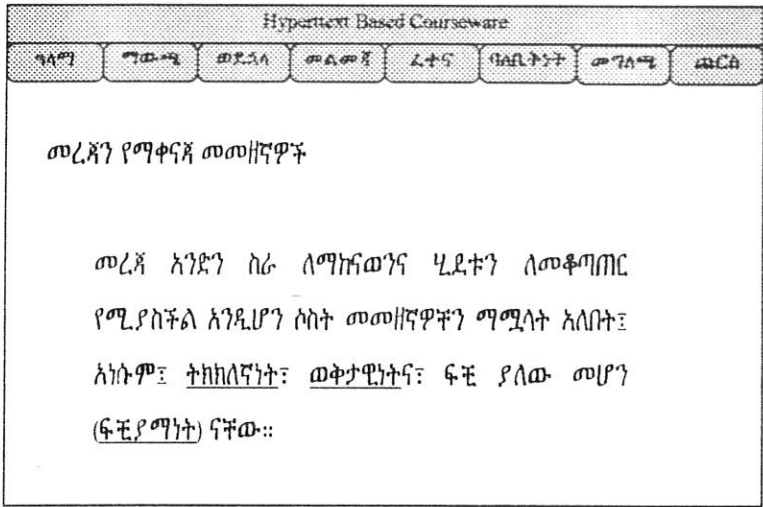


Figure 6.8 Module Overview Screen

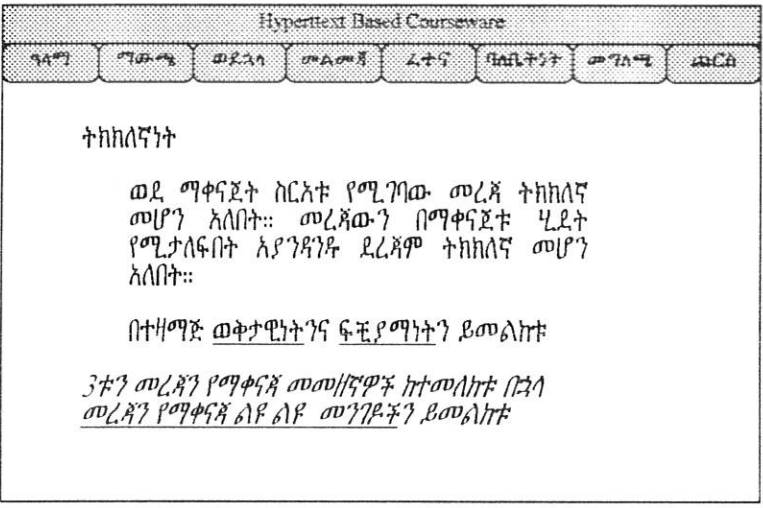


Figure 6.9 Module Elaboration with links to related items Screen

After the tutorial on each topic, the students are made to do practical exercises. Each exercise has a feedback on successful and unsuccessful answers. If a student scores below a certain level s/he is made to repeat the topic. Wrong answers are linked to the appropriate modules. Students securing a passing mark are made to revise some topics which they don't answer through the links of the wrong answers. See figures 6.10, 6.11, 6.12, 6.13, and 6.14 for the demonstration of these screens.

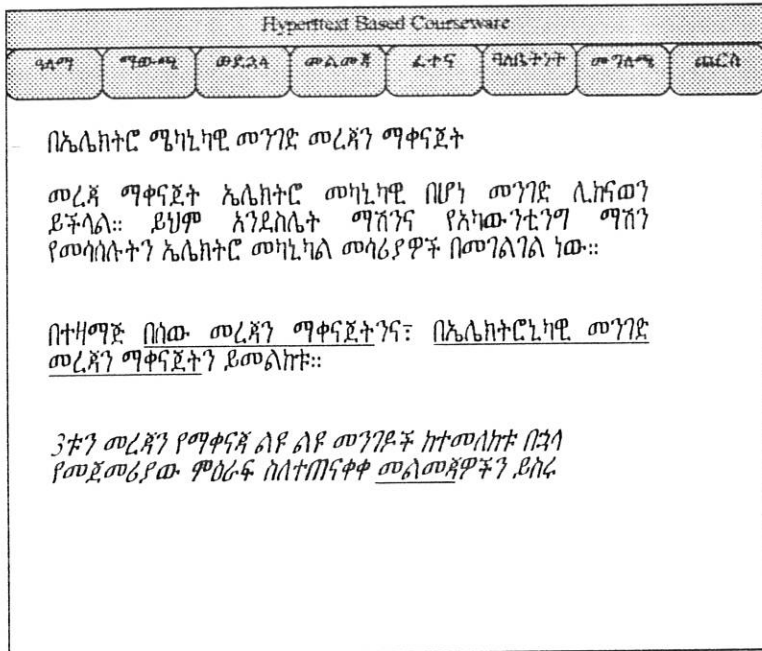


Figure 6.10. A last screen on a topic with a link to exercises.

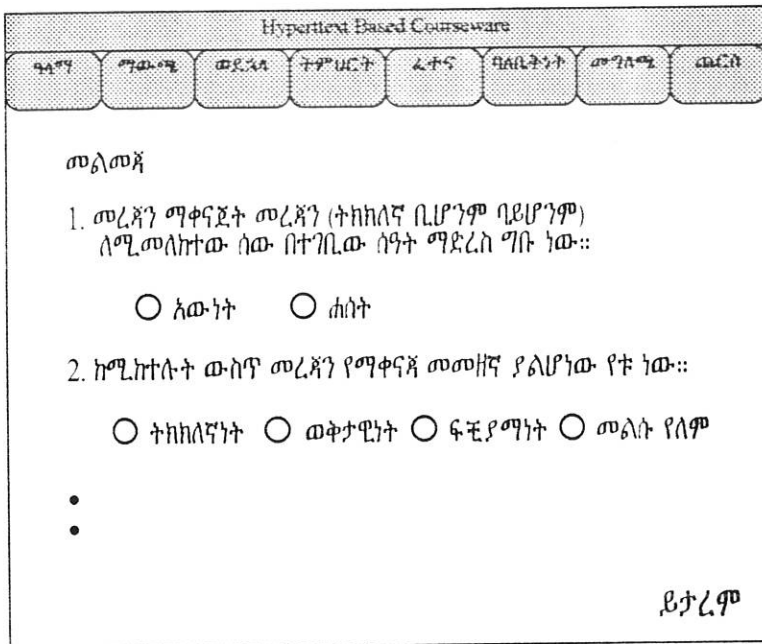


Figure 6.11 Exercise Screen

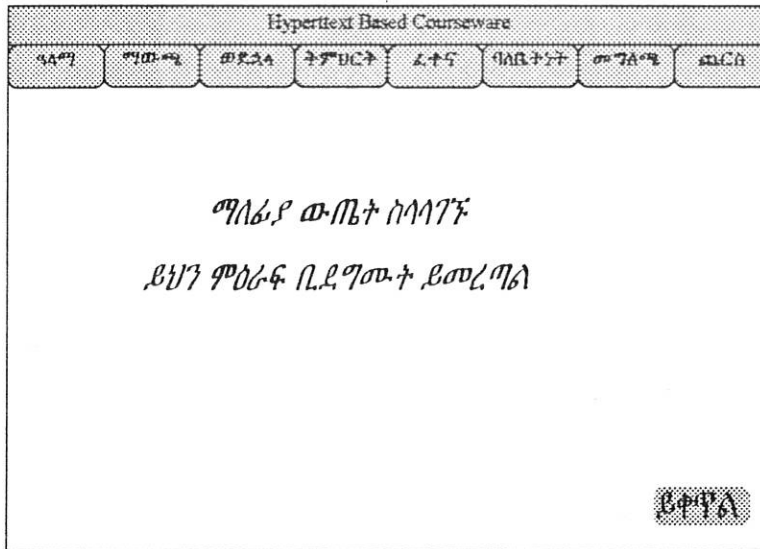


Figure 6.12 Information screen to repeat a topic (It is Animated and in Red color)



Figure 6.13 Information screen to proceed to the next lesson (It is Animated)

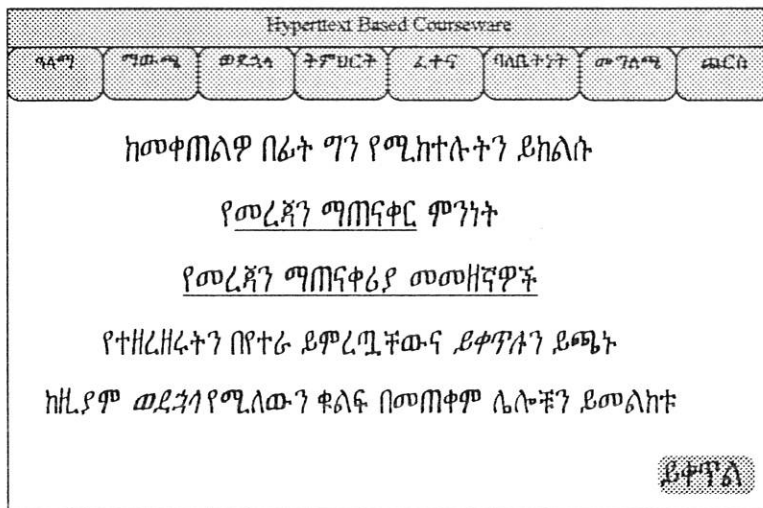





Figure 6.14 Information on incorrect answers with a link to the appropriate topics.

When learners finish the whole tutorial they are made to take a test. The test presents the test instruction screen first. They describe the rules of the test. Then a test sheet similar to the exercises screen is displayed. The test covers the whole tutorial topics and it is taken from a bank of examinations at NCIC. Learners can jump from question to question freely. They can also make any correction on their answers. Once satisfied with their answers, they can select the button  to see how much they scored. Like the exercise the test also evaluates their achievement based on the criteria indicated in the courseware objectives. Depending on their performance, they will be given a summary and concluding remarks including suggestions for future studies. The remark could be to repeat the course, to revise some topics or to end the courseware.

Help on how to use the courseware and explanations of each of the buttons is available by selecting the  button at any time. The information is also presented after the title screen for first time users. It can be made not to appear by selecting a check box.

Learners can leave the courseware at any time from any screen by clicking on the  button.

6.3 Implementation

The courseware is developed using KnowledgePro for Windows (Version 2.51) and runs in Microsoft Windows operating environment. It was developed on a 486 processor and 4 Mb of memory. It was not tested on other lower capacity computers capable of running Windows due to their unavailability. Therefore, the minimum system resource required could not be identified.

The current work shows the possibility of developing a hypertext based courseware in Amaharic by way of demonstrating the design and development of such a system. After incorporating the lessons and exercises for the remaining topics, the system can be fully implemented.

Currently SISA doesn't have KPWIN Gold or Borland C++ to make a distribution disk which doesn't want the KnowledgePro environment. Measures should be taken to acquire either of them. Therefore, this demonstration courseware needs the KnowledgePro environment to run in the background.

The courseware consists of one main program and three resource files for the lessons, exercises, and the test. These files can be installed in a user selected directory. A group window with the title Hypertext-Based Amaharic Courseware is created and an icon named IntroAm (Introduction in Amaharic) is added to the group window.

To install the courseware run the set up program using either of the following steps.

1. From the DOS prompt type **Win A:\setup**

or from the Windows program manager open the File menu and select Run.

type **A:\setup** and select OK.

2. Answer the questions provided by the installation program by typing or selecting buttons.

Once the installation program is completed the courseware can be started by double-clicking its icon.

CHAPTER SEVEN

CONCLUSION AND RECOMMENDATIONS

7.1 Conclusion

The prototype could not be tested due to lack of time. Subject to further testing of the prototype the following conclusion is made.

1. Currently in Ethiopia, computer utilization is felt in every economic sector. Business and government organizations are specifying computer skill as a prerequisite for employment. The need for training and retraining of staff in the field of computers is also strongly felt by many organizations.

The study has identified that there is a need for training on the basics of computers and that computer training institutions are mushrooming to respond to this need in Addis Ababa and in some of the major cities of the country.

2. It was also identified that except for those involved in the profession, the training result in the rest of the students is not satisfactory.
3. The research indicated that there is a shortage of properly trained instructors in the field. The emphasis given by training institutions is on the subject matter knowledge of instructors. The study also indicated that introducing Computer-Aided-Instruction into

the currently available computer training in the country could alleviate the problem. Besides, courseware can also be used by trainees to brush up their knowledge and to make what they acquire in training institutions sustainable.

4. The study has also shown the advantages of learning in one's own language. When learning is provided in one's own language the learning begins from what is already known. Among the languages in Ethiopia, Amaharic is identified to be the federal working language, medium of instruction in most schools, medium of communication in business, and having its own scripts. The development of the courseware in Amaharic is therefore advantageous.
5. Among the courses offered in computer training institutions, the course 'Introduction to computers' is found to be based entirely on lectures and manuals. The study has also indicated that this has a negative effect on students who come with the expectation of some interaction with computers. Due to this and other reasons it was found to be more suitable to introduce a courseware into the course 'Introduction to computers'.
6. The characters of the Amaharic language: letters, numbers, and punctuation marks, are altogether around 290. On the other hand, the Latin characters: letters, numbers, and punctuation marks, are not more than hundred. This fact has made almost all programming environments to have only one font, called the system font, to write programs inside their editors. Therefore, the available programming environments are not favourable for editing Amaharic programs.

In this study the screen interfaces are quickly developed and checked by running the program. When there were typing mistakes they were corrected. The quick prototyping approach has helped to test out the screen interfaces repeatedly.

The experience shows that, even though the strings of letters displayed in the program editor are meaningless, Amaharic applications can be developed using any application development environment as far as the typing of the Amaharic letters inside the editor is possible and the Amaharic fonts are accessed later at run time.

7. Reviews of literature in the area of courseware has indicated that any courseware should include four instructional phases: presenting information, guiding learning, practising, and assessing achievement.

It was also indicated that in a hypertext environment learners might have too much control, they might be lost in navigation, or might miss important information from the open nature of hypertext where no specific objectives or learning paths are specified.

Therefore, to reconcile the above two approaches were used. One approach is to guide novice learners through a fixed learning path by incorporating the nine events of instruction to make the courseware a complete instructional medium. The other approach is to give full responsibility for the learners for the success or failure of their learning by exploring the full feature of hypertext.

8. The review of courseware development models has helped to identify the necessary processes to be included in developing a courseware. Likewise the hypertext

development models reviewed have served the same purpose. This prototype courseware is developed by merging the processes from courseware development and hypertext development.

9. The courseware is developed to present information, guide the learning process, provide exercises for practising, and to assess the achievement of students. The exercise sessions in each lesson make sure whether a student is capable of going to the next lesson or not. If a student scores below a certain level he/she is made to go back and repeat the lesson. In addition, even if a passing score is found by a student the courseware will recommend the revision of some topics which are not correctly answered in the exercises. This will contribute for the complete understanding of a lesson by students.

The hypertext links between related materials enables students to see the relationship of the materials more clearly. This facilitates the retention of lessons by students.

10. From the experience gained in this study, it can be suggested that other courseware can be developed in Amaharic for other courses. In addition, Amaharic software packages developed so far do not have on-line help facilities. This problem can also be addressed by the current work.

7.2 Recommendations

7.2.1 For maturing the courseware

To mature the prototype to a full fledged courseware the following measures should have to be taken.

The main program is designed to access the lessons, exercises, and tests from external files. The hypertext links are marked in these external files. To incorporate the remaining lessons and exercises what remains is, therefore, to work on linking the information in a friendly environment which can handle the Amaharic scripts. That is, the editing of the complicated and somewhat confusing Amaharic interface in the program editor is taken care of.

In a related matter, if new technologies in computer field are introduced, appropriate topics can be added, chunked and linked in the word processing environment.

A user manual for the courseware should be prepared.

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding. In this regard test cases should be designed and the test results should be compared with expected results.

The courseware needs to be tested in two phases:

1. The first test is to check, with subject experts and students, whether the network created by buttons and links is really helpful, to find ways for further improvement of the courseware, and so on.

The course 'Introduction to computers' is given for at least two groups at any time at NCIC. This condition is favourable to perform the above mentioned test. The courseware can be installed for one group and the other group can proceed in the customary fashion.

At the end of the course the results can be compared and used to justify the usefulness, to find out places where improvement is necessary, or to disprove the findings of the study.

2. The second test is to check the performance of the courseware in real life testing environment among both novice and expert users. This test can be done by installing the courseware for two different groups. One copy can be installed for new students and the other copy can be installed for secretaries, and other computer users at NCIC which are not involved in teaching courses.

This test can also be used to set the average pass mark after evaluating the performance of different batches of students.

7.2.2 For further study

The current work deals with text only hypertexts. If a multimedia system and tools for authoring the different media elements is made available at SISA, this work can be extended to explore the advantage of hypermedia in education in Ethiopia.

Courses in computer literacy at all levels introduce students to the rudiments of programming and acquaint them with the purposes and functions of common computer applications, such as wordprocessing and spreadsheet programs. It is believed that beginners with computers may be overwhelmed by the initial complexity which a typical commercial software product presents (Leigh and Paz 1992). Instruction may proceed more

effectively if a suitably simplified training vehicle can be used to provide a conceptual model and an introduction to operational fundamentals before the use of the actual commercial software is attempted. Thus, the current work may be used as a stepping stone to develop such training products in Amaharic so that the intended objectives of national capacity building of some national institutions such as Ethiopian Science and Technology Commission will be met.

BIBLIOGRAPHY

1. Ahmed-Ouamer, Richard. 1993. AGEDI a computer assisted courseware engineering for industrial applications. In Proceedings of the IEEE International Conference on Systems Man and Cybernetics Part 2 (15) V (2): 540-545. New Jersey: IEEE service center.
2. Aktas, A. Z. 1987. Structured Analysis and Design of Information Systems. New Jersey: Prentice-Hall.
3. Arano, Takashi, et. al. 1993. An Object Oriented prototyping approach to system development. New Jersey: IEEE service center.
4. Barker, Philip and Manji, Karim. 1992. Computer based training: an institutional approach. Education and Computing. 8(3):229-237.
5. Barker, Philip. 1993. Exploring Hypermedia. London: Kogan Page Limited.
6. Berk, Emily and Devlin, Joseph. 1993. Hypertext/Hypermedia Handbook. New York: McGraw-Hill.
7. Binder, Carl. 1992. Hypertext: Design and Applications. Information Management. Auerbach Publications. Vol. II(4-06-90):1-11.

8. Bork, H. 1981. Learning with Computers. Bedford: Digital Press.
9. Bunderson, Victor C. 1981. Courseware: computer assisted instruction: a state of the art assessment. Orlando: Academic Press.
10. Choo, B. S. 1994. The BS8110 CAL Program for Teaching RC Design. Computing in Civil Engineering. 2:1663-1670.
11. Corenjo, J. G. 1994. Hypertext: Its use in the documentary treatment of data. Santiago, Chile: REDUCE-CIDE.
12. Creanor, Linda and Durndell, Helen. 1994. Teaching Information Handling Skills with Hypertext. Program. 28:349-365.
13. Emily, Jones C. 1978. Closing the Gap between Technology and Application. New York: EDUCOM.
14. Fiderio, Janet. 1988. A Grand Vision. Byte. 13(10):237-244.
15. Frisse, Mark F. and Cousins, Steve B. 1992. Models for Hypertext. Journal of the American Society for Information Science. 43(2):183-191.

16. Gagne, R. M., Briggs, L. J., and Wager, W. W. 1992. Principles of Instructional Design. 4th ed. Fort Worth: Harcourt Brace Jovanovich.
17. Getachew Worku. 1995. Factors Motivating Adult Trainees in Computer Training Centers in Addis Ababa. A Senior Essay. Addis Ababa University. Addis Ababa.
18. Gilbert, Philip. 1983. Software Design and Development. Chicago: Science Research Associates.
19. Hendrickson, Chris T., Janson, Bruce N., and Rossi, Mario. 1985. Applications and Potential of Micro-Computer Aided Instruction. Microcomputer within Transportation.
20. Issac, J. R. 1987. Human Resources Development and Computers in Education. In Proceedings of the 22nd Annual Convention of the Computer Society of India held at Bombay, 4-7 February. Edited by P Sadanandan and R Chandrasekar, 131-142. New Delhi: Tata McGrawHill Publishing Company Limited.
21. Jeffcoate, Judith. 1995. Multimedia in Practice: Technology and Application. Hertfordshire: Prentice Hall International (UK) Limited.

22. Jonassen, David H. and Wang, Sherwood. 1993. The Physics Tutor: Integrating Hypertext and Expert Systems. Journal of Educational Technology Systems. 22(1):19-28.
23. Jones, A. Russell and Steinberg, Esther R. 1990. Interface Design: An Educational Perspective. Journal of Systems Software. 11:199-204.
24. Jones, George W. 1990. Software Engineering. New York: John Wiley
25. Kenning, M. J. and Kenning M. M. 1988. An Introduction to Computer Assisted Language Teaching. Oxford: Oxford University Press.
26. Knowledge Garden Inc. 1991. KnowledgePro Windows Reference Manual. New York.
27. Knowledge Garden Inc. 1991. KnowledgePro Windows Users Manual. New York.
28. Lavin, Roberta Z. and Fechter, Sharon A. 1986. Online English for Computer Science. New York: McGraw-Hill.
29. Leigh, William and Paz, Noemi. 1992. The Development of Educational Replicas of Software Application Systems. Journal Educational Technology Systems. 21(1):71-78.

30. Littleford, Alan. 1991. Artificial Intelligence and Hypermedia. Hypertext/Hypermedia Handbook. Berk, Emily and Delvin, Joseph (eds). New York: McGraw-Hill.
31. Lucarella, D. and Zanzi, A. 1993. Information Retrieval from Hypertext: An approach using plausible inference. Information Processing and Management. 29(3):299-312.
32. Main, J. 1984. New Ways to Teach Workers What's New. Fortune Time Inc. Vol. 117. No. 7.:85-90.
33. Microsoft Corporation. 1993. Encarta 1994. (CD-ROM).
34. Morsy and Lamage. 1987. eds. Prospects: Quarterly Review of Education. Vol. XVII. No. 3. Paris: UNESCO.
35. Mulugeta Libsie. 1990. Introduction to Computers Training Manual. Addis Ababa: Ethiopian Science and Technology Commission, National Computer Centre.
36. National Computer and Information Centre. 1995. Facts Sheet. Addis Ababa. (Pamphlet).

37. Negussu Atnafu. 1993. Computer: A General Computer Awareness. Addis Ababa: CAROS Computer Service. (In Amaharic).
38. Parodi, G. and Ponta, D. 1993. Multimedia environment for computer based training in electronics and information technology. In Proceedings of the IFIP WG3.4 Working Conference on Computer mediated education of Information Technology Professionals and advanced end users (A 35): IFIP transaction A: Computer Science and Technology N A-35: North Holland: Elsevier Science Publishers.
39. Parsaye, Kamran et al. 1989. Intelligent Databases. New York: John Wiley & Sons.
40. Perez, Ernest. 1991. Tools for Authoring Hypertext. Hypertext/Hypermedia Handbook. Berk, Emily and Delvin, Joseph (eds). New York: McGraw-Hill.
41. Pistorius, M. C. et. al. 1992. Computer Assisted instruction development model for a computer concepts course. Computing and Control Engineering Journal. 3(1):13-18.
42. Rada, Roy. 1989. Writing and Reading Hypertext: An Overview. Journal of the American Society for Information Science. 40(3):164-171.

43. Rada, Roy. 1991. Focus on Links: A holistic view of hypertext. Knowledge Organization. 18(1):13-18.
44. Rahel Bekele. 1992. Information Technology Application in Higher Learning in Ethiopia: with special reference to computer application. M.Sc. Thesis Addis Ababa University. Addis Ababa.
45. Research & Planning Directorate. 1990. Computer Utilization Survey. the MANAGER. Vol 4. jan. 1990. Ethiopian Management Institute.
46. Rosenberg, Marc J. 1982. The ABC's of ISD. Trainig and Development Journal. (September):44-50.
47. Scholtz, Jean, et. al. 1993. Object Oriented Programming: The promise and the reality. North Holland: Elsevier Science Publishers.
48. Seyer, Philip. 1991. Understanding Hypertext: Concepts and Applications. PA: Windcrest Books/McGraw-Hill.
49. Shneiderman, Ben, Kreitzberg, Charles, and Berk, Emily. 1991. Editing to Structure a Reader's Experience. Hypertext/Hypermedia Handbook. Berk, Emily and Devlin, Joseph (eds). New York: McGraw-Hill.

50. Sirak G/Yohannes. 1988. Survey of the Development of Computer Use in Ethiopia. In Proceedings of the Seminar on Computer Applications and Informatics held at Addis Ababa, 25-27 April, sponsored by Ethiopian Science & Technology Commission. 31-72. Addis Ababa. Ethiopia.
51. Slatin, John M. 1991. Composing Hypertext: A Discussion for Writing Teachers. Hypertext/Hypermedia Handbook. Berk, Emily and Delvin, Joseph (eds). New York: McGraw-Hill.
52. Tagg, E. D. and Lovis, Frank. 1988. Computers in Education. Proceedings of the IFIP TC 3 1st European Conference on Computers in Education-ECCE 88. 24-29 July. North Holland: Elsevier Science Publishing Company.
53. Watts, Norman. 1981. A Dozen uses for computer in education. Journal of Educational Technology. (April):18-22.
54. Wong, Simon C-H.1993. Quick Prototyping of Educational Software: An Object Oriented Approach. Jornal of Educational Technology Systems. 22(2):155-172.
55. Yang, Chia-Shing, and Moore. David M. 1995. Designing Hypermedia Systems for Instruction. Jornal of Educational Technology Systems. 24(1):3-30.

APPENDIX

Program source code with sample lessons, exercises and tests

The program source code (HYPER1.KB) is given bellow.

(* A hypertext Based Computer Aided Instruction System in Amaharic

This system consists of one data and two source files for the lessons, exercises and a test. The source files are included in this main program.

Developed by: GIRMA DESSALEGN
MAY 1996
AAU/SISA
ADDIS ABABA

*)

```
#include exers.ckb
#include tests.ckb
title ().
```

```
(* ===== MAIN =====*)
```

```
setup ().
file is 'f:\home\girma\hyper\lessons.dat'.
getfile ().
show_window (?w1).
hlp_info is read_char ('f:\home\girma\hyper\help.inf').
close ('f:\home\girma\hyper\help.inf').
if ?hlp_info = f
    then help ().
objectives ().
```

```
(* ===== Lessons Data Handler ===== *)
```

```
topic getfile.
set_focus (?w1).
if ?file is []
    then exit ().
message is read (?file,'/', '/').
close (?file).
set_title (?wmain, 'Hypertext-Based Courseware in Amaharic').
top is string_replace (first (?message), '/').
list is [].
mark (?top).
end.
```

(* ===== Topics for the Buttons ===== *)

topic objectives.

set_focus (?w1).
mark ('የኮርሱ ዓላማ').
end. (*objectives*)

topic Content.

set_focus (?w1).
if last (?list) <> ?top then mark (?top).
end.

topic Back.

set_focus (?w1).
if list_length (?list) = 1
then mark (?top)
else
list is sublist (?list, 1, list_length (?list) - 1) and
item is last (?list) and
list is sublist (?list, 1, list_length (?list) - 1) and
(if ?item is help
then help ()
else mark (?item)).
end. (* Back *)

topic Exercise (section).

if get_text (?b4) <> 'መልመጃ'
then set_text (?b4, 'መልመጃ') and
hide_window (?wq) and
show_window (?w1) and
getfile ()
else set_text (?b4, 'ትምህርት') and
hide_window (?w1) and
show_window (?wq) and
instruction () and
Questions (?section) and
result ().
set_text (?b4, 'መልመጃ').
content ().
end. (* exercise selector *)

(* ===== INSTRUCTIONS ===== *)

topic instruction.

text ('#e ለሚቀርቡልዎ ጥያቄዎች ተስማሚ ነው ብለው ያመኑበትን ምርጫ
ከፊት ካለው ክብ ውስጥ ነጥብ በማድረግ ይምረጡ:::

መልሶቹን በአርግሜንት ከመረጡ በኋላ ይታረም የሚለውን ቁልፍ በመምረጥ
ውጤትዎን ይመልከቱ:::

ይታረም የሚለውን ቁልፍ ከመምረጥዎ በፊት ማንኛውንም መልስ ለማስተካከል
ይችላሉ። ለማስተካከል አዲሱን መልስዎን ይምረጡ። ነገሩ ወደምረጫዎ
ትህድልዎታለች።).

button ('ይቀጥል',continue, 10, 23).
wait ().
end. (*instruction *)

(* ===== EXERCISES ON EACH LESSON ===== *)

topic 'መልመጃ አንድ'. (* on lesson one*)
exercise('መግቢያ').
end.

topic 'መልመጃ ሁለት'. (* on lesson two*)
exercise('ስለ ኮምፒዩተሮች').
end.

topic 'መልመጃ ሶስት'. (* on lesson three*)
exercise('ኮምፒዩተር ሀረጎች').
end.

topic 'መልመጃ አራት'. (* on lesson four*)
exercise('ኮምፒዩተር ሶፍትዌር').
end.

topic 'መልመጃ አምስት'. (* on lesson five*)
exercise('ኮምፒዩተር ኮሙኒኬሽን').
end.

topic exercises. (* on a selected lesson*)
exercise ().
end. (* ===== end exercises block ===== *)

topic test.
hide_window (?w1).
show_window (?wq).
instruction ().
test_quest ().
feedback ().
end. (* test *)

topic credits.
mark('ባለቤትነት').
end. (*credits*)

topic Help.
set_focus (?w1).
if last (?list) is 'Help'

```

then exit ().
list gets Help.
show_window (?wq).
text ('#e #rb የማስተማሪያው አጠቃቀም መግለጫ#rd

```

አረንጓዴው ፅሁፍ ሀይማኖታዊ ማሳሰቢያ።።። ሲመረጡት ተጨማሪ መረጃ ያመጣልዎታል።።።
 ለመምረጥ የሚውሉ ቀንቀሮች ወይንም በመውሰድ ማውሰን ይጫኑ ወይም ቦታ መወሰንን
 ተጠቅመው ቀንቀሮችን ከአንድ ሀይማኖታዊ ማሳሰቢያ ያንቀሳቅሱና ለመምረጥ ይግባን ይጫኑ።።።

```

#mይህ የሀይማኖታዊ ማሳሰቢያ ነው#m አስኪ ይምረጡት

```

```

የማስተማሪያውን ዓላማ ለማየት #bgray ዓላማ #d የሚለውን ቁልፍ ይምረጡ።።።
ወደ መጀመሪያው ስኬት ለመሄድ #bgray ማውጫ #d የሚለውን ቁልፍ ይምረጡ።።።
አንድ ስኬት ወደ ኋላ ለመመለስ #bgray ወደኋላ #d የሚለውን ቁልፍ ይምረጡ።።።
መልመኝዎችን ለመስራት #bgray መልመኝ #d የሚለውን ቁልፍ ይምረጡ።።።
ፈተና ለመስራት #bgray ፈተና #d የሚለውን ቁልፍ ይምረጡ።።።
ይህን መግለጫ ለማየት #bgray መግለጫ #d የሚለውን ቁልፍ ይምረጡ።።።
ከማስተማሪያው ለመውጣት #bgray ጨርስ #d የሚለውን ቁልፍ ይምረጡ።።።)

```

topic 'ይህ የሀይማኖታዊ ማሳሰቢያ ነው'.

```

text ('#x1 #y23 #fred #bcyan ጥሩ መርጠዎል፤ አንግሊክ ሌሎችንም አንዲሁ ይምረጡ #d').
wait(,2).
location is search_text (?wq, 'ጥሩ መርጠዎል፤ አንግሊክ ሌሎችንም አንዲሁ ይምረጡ').
select_text (?wq, ?location).
wait(,2).
delete_selected_text (?wq, ?location).
end. (* a hypertext example *)

```

```

set_display_pos (1,24).
hlp_info is read_char ('f:\home\girma\hyper\help.inf').
close ('f:\home\girma\hyper\help.inf').
cb is check_box ('ይህ መግለጫ መጀመሪያ ላይ አይሞጣ',checked,,?hlp_info).

```

```

topic checked.
new_file ('f:\home\girma\hyper\help.inf').
if get_check_box (?cb)
then write ('f:\home\girma\hyper\help.inf,t)
else write ('f:\home\girma\hyper\help.inf,f).
close ('f:\home\girma\hyper\help.inf').
end. (*checked*)

```

```

button ('ይቀጥል', back, 55, 23).

```

wait ().

end. (*Help*)

topic Quit.

exit_kp().

end. (* Quit *)

(* ===== TOPIC TO CLEAN UP ON EXIT ===== *)

topic Finish.

clear ().

end.

(* ===== MARK - HYPERTEXT HANDLER ===== *)

topic mark (item).

list gets?item.

displayText is read (?file, concat ('//',?item), '//') .

if ?displayText is number_to_char (26)

then displayText is '

ለጊዜው መረጃ የለውም !

set_text (?w1,[#rb,?item,#rd, ?displayText]).

set_file_pos (?file,0,beginning).

end. (*mark*)

(* ===== SELECT FONTS ===== *)

topic en.

use_font (?engFont).

end.

topic b.

use_font (?titleFont).

end.

topic d.

use_font (?mainFont).

end.

topic i.

use_font (?italicFont).

end.

(* ===== SETUP ===== *)

topic setup.

```
wMain is window (Finish,3,1,88,30, ,[popup, thinFrame,TitleBar], , , , ).
wB is window ( , 1,1,100,2, ,[child, disabled, siblings, showChildren], ?wmain, , gray).
buttonFont is create_char_font ( [1,1.5,400,'F','F','F',0,1,34,'AGF - Zemen']).
use_font (?buttonFont,CONTROL).
show_window (?wB).
b1 is button ('ዓላማ',objectives,1,1,11).
b2 is button ('ማውጫ',content,12,1,11).
b3 is button ('ወደኋላ',back,23,1,11).
b4 is button ('መልመሻ',exercises,34,1,11).
b5 is button ('ፈተና',test,45,1,11).
b6 is button ('ባለቤትነት',credits,56,1,11).
b7 is button ('መግለጫ',help,67,1,11).
b8 is button ('ጨርስ',quit,78,1,11).
enable_window (?wb).
w1 is window ( , 10,4, element ( window_info (?wMain), 10),
  element ( window_info (?wMain), 11)-4, , [child, siblings, showChildren] ,?wmain).
wq is window ( ,10,4, 78, 26, , [child, showchildren, siblings], ?wMain).
engFont is create_char_font ( [1,1.25,400,'F','F','F',0,1,34,'Arial']).
mainFont is create_char_font ( [1,1.25,400,'F','F','F',0,1,34,'AGF - Rejim']).
italicFont is create_char_font ( [1,1.25,400,'T','F','F',0,1,34,'AGF - Rejim']).
hyperFont is create_char_font ( [1,1.25,400,'F','T','F',0,1,34,'AGF - Rejim']).
titleFont is create_char_font ( [1,1.5,400,'T','F','F',0,1,34,'AGF - Rejim']).
use_font (?mainFont).
if last (system_info ()) > 2
  then :color is green.
  else :color is black.
hyper_display (?color,,?hyperFont).
show_window (?wmain).
list is [].
end.
```

```
(*=====TITLE SCREEN=====*)
```

```
topic title.
titlescr is load_bitmap('f:\home\girma\hyper\titlescr.bmp').
window ( ,1,1,102,30,'Hypertext-Based Courseware in Amaharic').
bitmap (?titlescr,1,1,90,27).
zemen is create_char_font ([1,1.5,400,f,f,f,0,1,34,'AGF - Zemen']).
use_font (?zemen, control).
button ('ይቀጥል',Continue,43,25).
wait ( ,15).
close_window ().
delete_bitmap (?titlescr).
delete_font (?zemen).
end.
```

The data for the first lesson (LESSONS.DAT) is given bellow.

ሀሀማውጫ

#mመግቢያ#m

#mስለ ኮምፒዩተርች#m

#mhምፒዩተር ሀርድዌር#m

#mhምፒዩተር ሶፍትዌር#m

#mhምፒዩተር ኮሙኒኬሽን#m

#mያርዕስት ማውጫ#m

ሀሀየኮርሱ ዓላማ

ይህን ኮርስ ካጠናቀቃችሁ በኋላ፣ የሚከተሉትን ማከናወን ይጠበቅባችኋል፤

1. ስለ መረጃን ማቀናጀት መሰረታዊ ክፍሎች አና ለምን አንደሚያስፈልግ ማወቅ
2. በኮምፒዩተር መስክ ውስጥ ስላሉ ሃሳቦችና ቃላት ማወቅ
3. ኮምፒዩተር ሊያደርጉ የሚችሉትንና የማይችሉትን ሊይቶ መረዳት
4. የኮምፒዩተርን ልዩ ልዩ ክፍሎች (ሀርድዌርና ሶፍትዌር) ማወቅ
5. ስለ መረብ መሰረታዊ ሃሳቦች ማወቅ

#fred #ri ወደ #mማውጫ#mው ይሂዱ #d #rd

ሀሀያርዕስት ማውጫ

#mመረጃን ማቀናጀት#m #t#t#mመረጃን የማቀናጃ ልዩ ልዩ መንገዶች#m

#mመረጃን የማቀናጃ መመዘኛዎች#m#t#mማህደረ ትውስታ#m

#mሲፒዩ#m#t#t#mጠሰው መረጃን ማቀናጀት#m

#mበኤሌክትሮ ሜካኒካዊ መንገድ መረጃን ማቀናጀት#m

#mበኤሌክትሮኒካዊ መንገድ መረጃን ማቀናጀት#m

#mቢት#m#t#t#mባስ ሲስተም#m

#mባይት#m#t#t#mባይነሪ ኮድድ ዲሲማል#m

#mአሜሪካን ስታንዳርድ ኮድ ፎር ኢንፎርሜሽን ኢንተርፕሪንድ#m

#mhምፒዩተር#m#t#t#mhድ የማድረጊያ ዘዴዎች#m

#fred ቀዳሞቹን ለማየት #mያርስት ማውጫ ቀጣይ#mን ይምረጡ

ሀሀያርስት ማውጫ ቀጣይ

#mወርድ#m#t#t#mውጫዊ ማህደረ ትውስታ#m

#mውጣት ክፍል#m#t#mየኮምፒዩተር የአቆጣጠር ስርዓቶች#m

#mግባት ክፍል#m

#fred #ri ወደ #mማውጫ#mው ይሂዱ #d #rd

ሀሀባለቤትነት

ይህ ፕሮግራም በዶ/ር ጂ. ጂ. ችውድሪና በአቶ ነጋ አለማየሁ አማካሪነት በአቶ ግርማ ደሳለን የተዘጋጀ ነው። አቶ ግርማ ደሳለን በአዲስ አበባ ዩኒቨርሲቲ በአፍሪካ የኢንፎርሜሽን ጥናት ት/ቤት የድህረ ምረቃ ተማሪ ነው።

የኮርሱ ይዘት በብሔራዊ የኮምፒዩተርና ኢንፎርሜሽን ማዕከል ከሚሰጠው #ren Introduction to Computers #rd ኮርስ የተወሰደ ነው።

ግንቦት 1988 ዓ.ም
አዲስ አበባ

ሀሀመግቢያ

ይህን ምዕራፍ ካጠናቀቃችሁ በኋላ፤

የ#mመረጃን ማቀናጀት#mን ምንነት፣

#mመረጃን የማቀናጃ መመዘኛዎች#mንና፣

#mመረጃን የማቀናጃ ልዩ ልዩ መንገዶች#mን

ማወቅ ይጠበቅባችኋል።

ሀሀመረጃን ማቀናጀት

መረጃን ማቀናጀት ምንድነው#ren ? #rd

መረጃን ማቀናጀት መረጃን የመቀበል፣ ቀደም ብሎ በተደነገገ ህግ መሠረት መረጃን የመለወጥ፣ አና ውጤትን የማግኘት አንቅስቃሴ ነው።

መረጃን ማቀናጀት ትክክለኛ መረጃን በተገቢው ጊዜ ለሚመለከተው (ትክክለኛው) ሰው አንዲያገኘው ማድረግ ጠቀላላ ግቡ የሆነ ሂደት ነው።

ለተጨማሪ ማብራሪያ #mመረጃን ማቀናጀት ማብራሪያ#mን ይመልከቱ።

#fred #ri ቀጥለው #mመረጃን የማቀናጃ መመዘኛዎች#mን ይመልከቱ #d #rd

ሀሀመረጃን ማቀናጀት ማብራሪያ

መረጃን ማቀናጀት ትክክለኛ መረጃን በተገቢው ጊዜ ለሚመለከተው (ትክክለኛው) ሰው አንዲያገኘው ማድረግ ጠቀላላ ግቡ የሆነ ሂደት ነው።

የማቀናጀቱ ሂደት ተሰብስቦ ካለ መረጃ ውስጥ ለሚመለከተው አካል ፍቺ ያለው መልዕክት ማቅረብ (ማስተላለፍ) አንዲቻል ተገቢውን (አስፈላጊውን) አውነታ መምረጥንና ማጠናቀርን ያካትታል።

አንድ አምራች ድርጅት አያሌ ቅርንጫፎች፣ ሠራተኞች ከፍተኛ ምርትና ተወዳሪዎች ሲኖሩትና አንዲሁም ሳይንሳዊ ምርምርን ለማድረግ መረጃን ማቀናጀት አጅግ አስፈላጊ ጉዳይ ይሆናል። በምርምር ስራ፣ በማምረት ሂደት፣ በገበያ፣ በሰራተኞች ጉዳይ ወዘተ. አስቸኳይ ውሳኔ ለመስጠት የሚያስችል ትክክለኛ መረጃ በወቅቱ ማግኘት የሚችሉ ናቸው ሳይወድቁ መዘለቅን የቻሉ።

#fred #ri ቀጥለው #መረጃን የማቀናጃ መመዘኛዎች#mን ይመልከቱ #d #rd

ሀሀመረጃን የማቀናጃ መመዘኛዎች

መረጃ አንድን ስራ ለማከናወንና ሂደቱን ለመቆጣጠር የሚያስችል አንዲሆን ሶስት መመዘኛዎችን ማሟላት አለበት፤ አነሱም፤ #mትክክለኛነት#m፣ #mወቅታዊነት#m፣ ፍቺ ያለው መሆን (#mፍቺያማነት#m) ናቸው።

ሀሀትክክለኛነት

አንድ መረጃን የማቀናጃ መመዘኛ ትክክለኛነት ነው። ወደ ማቀናጀት ስርአቱ የሚገባው መረጃ ትክክለኛ መሆን አለበት። መረጃውን በማቀናጀቱ ሂደት የሚታለፍበት አያንዳንድ ደረጃም ትክክለኛ መሆን አለበት።

በተዛማጅ #mወቅታዊነት#mንና #mፍቺያማነት#mን ይመልከቱ

#fred #ri 3ቱን መረጃን የማቀናጃ መመዘኛዎች ከተመለከቱ በኋላ #መረጃን የማቀናጃ ልዩ ልዩ መንገዶች#mን ይመልከቱ #d #rd

ሀሀወቅታዊነት

አንድ መረጃን የማቀናጃ መመዘኛ ወቅታዊነት ነው። ዘግይቶ የቀረበ ትክክለኛ መረጃ ጠቅላላውን መረጃ ያለማግኘትን ያህል ዋጋቢ ነው። (ይህም ማለት የመረጃ ማቀናጃ ስርዓት ፈጣን አንዲሆን ይጠይቃል)።

በተዛማጅ #mትክክለኛነት#mንና #mፍቺያማነት#mን ይመልከቱ

#fred #ri 3ቱን መረጃን የማቀናጃ መመዘኛዎች ከተመለከቱ በኋላ #መረጃን የማቀናጃ ልዩ ልዩ መንገዶች#mን ይመልከቱ #d #rd

ሀሀፍቺያማነት

አንድ መረጃን የማቀናጃ መመዘኛ ፍቺያማነት ነው። ከመረጃ ማቀናጃ ሂደት የሚገኘው መረጃ ለተጠቃሚዎች ፍቺ ያለው መሆን አለበት፤ ማለትም

ሀ) መረጃው ለተጠቃሚዎቹ ፍሎጎት የሚመጥንና ተገቢ መሆን አለበት

ለ) የሚቀርበው ሪፖርት ያለችግር የሚገነዘቡት መሆን አለበት። የሪፖርቱ ቅርፅም ሆነ ይዘት በቀላሉ የሚነበብና የሚገባ መሆን አለበት።

በተዛማጅ #mትክክለኛነት#mንና #mወቅታዊነት#mን ይመልከቱ

#fred #ri 3ቱን መረጃን የማቀናጃ መመዘኛዎች ከተመለከቱ በኋላ #መረጃን የማቀናጃ ልዩ ልዩ መንገዶች#mን ይመልከቱ #d #rd

ሀሀመረጃን የማቀናጃ ልዩ ልዩ መንገዶች

ሁሉም ዓይነት ስራ መረጃን የማቀናጀት ፍላጎቶች አሉት፤ በዚህም የተነሳ ከአንድህ ፍላጎቶቹ ጋር የሚጣጣም መረጃን የማቀናጀት ዘዴ መምረጥ አለበት። አንድህም ዘዴዎች #መሰረድ መረጃን ማቀናጀት#m፣ #መሰረድህክትሮ ሜካኒካዊ መንገድ መረጃን ማቀናጀት#m፣ #መሰረድህክትሮህክትሮህክትሮ መንገድ መረጃን ማቀናጀት#m ናቸው።

ለተጨማሪ ማብራሪያ #መረጃን የማቀናጀት ልዩ ልዩ መንገዶች ማብራሪያ#mን ይመልከቱ

ሀሀመረጃን የማቀናጀት ልዩ ልዩ መንገዶች ማብራሪያ

ሁሉም ዓይነት ስራ መረጃን የማቀናጀት ፍላጎቶች አሉት፤ በዚህም ታተነሳ ከአንድህ ፍላጎቶቹ ጋር የሚጣጣም መረጃን የማቀናጀት ዘዴ መምረጥ አለበት።

የመረጃ ማቀናጀት አይነትና መሳሪያውም ቢሆን መመሪያ ያለበት መረጃ ማቀናጀትን አስፈላጊ ያደረጉት ችግሮችና ለመፍታት የሚጠይቀው ወጪ በጥልቀት ተተንተኖ በተንተኖ ላይ በመመስረት ነው። አያንዳንዱ ሁኔታ ልዩ ባሕሪያቱን በሚያመለክቱ መንገድ መመዘን አለበት።

#መሰረድ መረጃን ማቀናጀት#m፣
#መሰረድህክትሮ ሜካኒካዊ መንገድ መረጃን ማቀናጀት#mን፣
#መሰረድህክትሮህክትሮህክትሮ መንገድ መረጃን ማቀናጀት#mን ይመልከቱ።

#fred #ri 3ቱን መረጃን የማቀናጀት ልዩ ልዩ መንገዶች ከተመለከቱ በኋላ የመጀመሪያው ምዕራፍ ስለተጠናቀቀ #መመልመጃ አንድ#mን ይስሩ #d #rd

ሀሀበሰው መረጃን ማቀናጀት

መረጃን የማቀናጀት ስራ በሰው ሊከናወን ይችላል። በዚህ ጊዜ አስምሮ ወረቀትና አርሳስ ዋነኞቹ መሳሪያዎች ናቸው።

ለተጨማሪ ማብራሪያ #መሰረድ መረጃን ማቀናጀት ማብራሪያ#mን ይመልከቱ

ካስፈለገ #መረጃን የማቀናጀት ልዩ ልዩ መንገዶች ማብራሪያ#mን ይመልከቱ

በተዛማጅ #መሰረድህክትሮ ሜካኒካዊ መንገድ መረጃን ማቀናጀት#mን፣ #መሰረድህክትሮህክትሮ መንገድ መረጃን ማቀናጀት#mን ይመልከቱ።

#fred #ri 3ቱን መረጃን የማቀናጀት ልዩ ልዩ መንገዶች ከተመለከቱ በኋላ የመጀመሪያው ምዕራፍ ስለተጠናቀቀ #መመልመጃ አንድ#mን ይስሩ #d #rd

ሀሀበሰው መረጃን ማቀናጀት ማብራሪያ

አንዳራዊ በሆነ መንገድ መጠኑ አነስተኛ የሆነ ስራን የሚሰራ አነስተኛ ድርጅት (ካምፓኒ) በሰው የሚከናወነውን የመረጃ ማቀናጀት ዘዴ በቂ ሆኖ ያገኘዋል።

መረጃን በሰው የማቀናጀት ዘዴ ሰው ሊሳሳት በመቻሉ ተመሳሳይ ስራን የመደጋገም ችግር በማጋጠሙ፣ የሰውተኞች ምርታመነት ውስን በመሆኑ ለመቆጣጠር አመቺ ባለመሆኑና የአምነት መጉደል ችግር ሊከሰት በመቻሉ አያሌ ጉዳዮች ያጋጥሙታል። ይሁን እንጂ የመረጃው መጠን አነስተኛ ከሆነ ወይም ሪፖርቱ የሚቀርብበት የጊዜ ገደብ መረጃውን ለማቀናጀት ከሚያስፈልገው ጊዜ ጋር ሲነፃፀር ሰው ከሆነ ወይም የውጤቱ ተክክለኝነት የግድ ካልሆነ ወይም የበጀቱ መጠን የመረጃ ማቀናጀት መሳሪያዎችን ለመግዛት የማይፈቀድ ከሆነ ጥሩው ምርጫ በሰው መረጃን የማቀናጀት ዘዴ ይሆናል።

በተዛማጅ #መሰረድህክትሮ ሜካኒካዊ መንገድ መረጃን ማቀናጀት#mን፣ #መሰረድህክትሮህክትሮ መንገድ መረጃን ማቀናጀት#mን ይመልከቱ።

ካስፈለገ #መረጃን የማቀናጃ ልዩ ልዩ መንገዶች ማብራሪያ#mን ይመልከቱ

#fred #ri 3ቱን መረጃን የማቀናጃ ልዩ ልዩ መንገዶች ከተመለከቱ በኋላ የመጀመሪያው ምዕራፍ ስለተጠናቀቀ #መልመጃ አንድ#mን ይስሩ #d #rd

ሀሀበኤሌክትሮ ሜካኒካዊ መንገድ መረጃን ማቀናጀት

መረጃ ማቀናጀት ኤሌክትሮ መካኒካዊ በሆነ መንገድ ሊከናወን ይችላል።። ይህም አንደኛው ማሸንፍ የአካውንቲንግ ማሸንፍ የመሳሰሉትን ኤሌክትሮ መካኒካል መሳሪያዎች በመገልገል ነው።።

ለተጨማሪ ማብራሪያ

#መበኤሌክትሮ ሜካኒካዊ መንገድ መረጃን ማቀናጀት ማብራሪያ#mን ይመልከቱ

ካስፈለገ #መረጃን የማቀናጃ ልዩ ልዩ መንገዶች ማብራሪያ#mን ይመልከቱ

በተዛማጅ #መበሰው መረጃን ማቀናጀት#m ንፍ፣

#መበኤሌክትሮኒካዊ መንገድ መረጃን ማቀናጀት#mን ይመልከቱ።።

#fred #ri 3ቱን መረጃን የማቀናጃ ልዩ ልዩ መንገዶች ከተመለከቱ በኋላ የመጀመሪያው ምዕራፍ ስለተጠናቀቀ #መልመጃ አንድ#mን ይስሩ #d #rd

ሀሀበኤሌክትሮ ሜካኒካዊ መንገድ መረጃን ማቀናጀት ማብራሪያ

የአንድ ድርጅት ስራ አያደገ ሲሄድ ከመረጃ ማቀናጃ ስርዓቱ ትክክለኛ፣ የተሟላ፣ ፍቺ ያለው መረጃ የማግኘት ፍላጎቱም አብሮ ያድጋል።። በዚህ ጊዜ መዋዕል ንዋይን ለሜካኒካልና ኤሌክትሮኒክ መሳሪያዎች መመደብ የግድና መረጃ የማቀናጀትንም ችግር በዝቅተኛ ወጪ የመፍቻው መንገድ ይሆናል።።

ካስፈለገ #መረጃን የማቀናጃ ልዩ ልዩ መንገዶች ማብራሪያ#mን ይመልከቱ

በተዛማጅ #መበሰው መረጃን ማቀናጀት#m ንፍ፣

#መበኤሌክትሮኒካዊ መንገድ መረጃን ማቀናጀት#mን ይመልከቱ።።

#fred #ri 3ቱን መረጃን የማቀናጃ ልዩ ልዩ መንገዶች ከተመለከቱ በኋላ የመጀመሪያው ምዕራፍ ስለተጠናቀቀ #መልመጃ አንድ#mን ይስሩ #d #rd

ሀሀበኤሌክትሮኒካዊ መንገድ መረጃን ማቀናጀት

መረጃን ማቀናጀት ኤሌክትሮኒካዊ በሆነ መንገድ የኮምፒዩተር ስርዓትን በመጠቀም ሊከናወን ይችላል።።

ለመረጃ ማቀናጀት ስራ የኤሌክትሮኒክ መሳሪያዎች አገልግሎት ላይ በሚውሉበት ጊዜ ከፍተኛ የሆነ ፍጥነትን፣ ትክክለኛነትንና፣ ውጤታማነትን ያስገኛሉ።።

ካስፈለገ #መረጃን የማቀናጃ ልዩ ልዩ መንገዶች ማብራሪያ#mን ይመልከቱ

በተዛማጅ #መበሰው መረጃን ማቀናጀት#m ንፍ፣

#መበኤሌክትሮ ሜካኒካዊ መንገድ መረጃን ማቀናጀት#mን ይመልከቱ።።

#fred #ri 3ቱን መረጃን የማቀናጃ ልዩ ልዩ መንገዶች ከተመለከቱ በኋላ የመጀመሪያው ምዕራፍ ስለተጠናቀቀ #መልመጃ አንድ#mን ይስሩ #d #rd

Source code for a sample exercise (EXERS.KB) is given below.

count is 0.

topic questions (sect).

if ?sect = []

then text ('#e መልመጃ ሊሰፈርበት የሚፈልጉትን ምዕራፍ ይምረጡ' #n#n) and

sec is radio_button ([['መግቢያ', 3, 3], ['ስለ ኮምፒዩተሮች', 3, 5],

['ኮምፒዩተር ሀርድዌር', 3, 7], ['ኮምፒዩተር ሶፍትዌር', 3, 9], ['ኮምፒዩተር ኮሙኒኬሽን', 3, 11]) and

button ('ይቀጥል', continue, 7, 23) and

wait () and

sect is radio_value(?sect).

do (?sect).

topic 'መግቢያ'. (* Exercise on lesson one *)

wrong is [].

text ('#e 1. መረጃን ማቀናጀት መረጃን (ትክክለኛ ቢሆንም ባይሆንም) ለሚመለከተው ሰው በተገቢው ሰዓት ማድረስ ግቡ ነው::: #n).

q1 is radio_button ([['አውነት', 3, 3], ['ሐሰት', 13, 3]]).

text ('#x1 #y5 2. ከሚከተሉት ውስጥ መረጃን የማቀናጃ መመዘኛ ያልሆነው የቱ ነው #ren?#rd).

q2 is radio_button([['ትክክለኛነት', 3, 6], ['ወቅታዊነት'], ['ፍቺያማነት'], ['መልሱ የለም']]).

text ('#x1 #y8 3. ከፍተኛ ፍጥነት፣ ትክክለኛነትና ውጤታማነት የሚያስገኘው መረጃን የማቀናጃ መንገድ የትኛው ነው፡፡ ብቅቻላ፡፡ ብቁ).

q3 is radio_button ([['ብሰው', 3, 9], ['በኤሌክትሮኒካዊ'], ['በኤሌክትሮ ሜካኒካዊ'], ['መልሱ የለም']]).

text ('#x1 #y11 4. መረጃን በሰው ማቀናጀት ሊያጋጥሙት ከሚችሉት ችግሮች ውስጥ).

q4 is radio_button ([['ሰው ሊሳሳት መቻሉ', 3, 12], ['ተመሳሳይ ስራን መደጋገሙ', 35, 12], ['ዕምነትን ማጉደል ሊከስት መቻሉ', 3, 13, 2], ['ሁሉም መልስ ነው']]).

text ('#x1 #y15 5. መረጃን የማቀናጃ መንገድ ወይም መሣሪያ ለመምረጥ መረጃ ማቀናጀቱን አስፈላጊ ያደረጉትን ችግሮች ማወቁ ብቻ በቂ ነው:::').

q5 is radio_button ([['አውነት', 3, 17], ['ሐሰት']]).

button ('ይታረም', continue, 7, 23).

wait ()

if radio_value (?q1) is 'ሐሰት'

then count = ?count + 1

else count = ?count and

wrong gets '#m መረጃን ማቀናጀት #m'.

if radio_value (?q2) is 'መልሱ የለም'

then count = ?count + 1

else count = ?count and

wrong gets '#m መረጃን የማቀናጃ መመዘኛዎች #m'.

```

if radio_value (?q3) is 'በኤሌክትሮኒካዊ'
  then count = ?count + 1
  else count = ?count and
  wrong gets '#mበኤሌክትሮኒካዊ መንገድ መረጃን ማቀናጀት#m'.

```

```

if radio_value (?q4) is 'ሁሉም መልስ ነው'
  then count = ?count + 1
  else count = ?count and
  wrong gets '#mበሰው መረጃን ማቀናጀት ማብራሪያ#m'.

```

```

if radio_value (?q5) is 'አስት'
  then count = ?count + 1
  else count = ?count and
  wrong gets '#mመረጃን የማቀናጃ ልዩ ልዩ መንገዶች#m'.

```

end. (* Exercise on lesson one *)

```

topic radio_value (handles).
  selected is element (?handles,
    where (get_radio_button (?handles), t)).
  radio_value is get_text (?selected).
end.

```

```

topic 'ስለ ኮምፒዩተሮች'.
  text ('#e #x1 #y3 ስለኮምፒዩተሮች መልመጃ').
  button ('ይቀጥል','go_on',7,23).
end.
end. (* questions *)

```

```

topic result.
  if ?count < 3
  then repeatit ()
  else proceed () and
    button ('ይቀጥል',continue,7,23) and
    wait ()and
    wrongans ().
end. (* result *)

```

```

topic repeatit.
  text ('#e #fred #x1      ማለፊያ ውጤት ስላላገኙ
    ይህን ሞዕራፍ ቢደግሙት ይመረጣል #d').
  animate ().
  button ('ይቀጥል',continue,7,23).
  wait ().
  'go_on' ().
end.

```

topic proceed.

```
text ('#e ማለፊያ ውጤት አግኝተዋል።።።  
ወደሚቀጥለው ምዕራፍ ይለፉ').  
animate ().  
end.
```

topic animate.

```
r = 4.  
while ?r < 24  
    then move_window (?wq,?r,5) and  
        wait (,0.1) and  
        r = ?r + 2.  
end.
```

topic wrongans.

```
if ?wrong <> [] then  
    text ('#e ከመቀጠልዎ በፊት የሚከተሉትን ይክልሉ',#n#n, ?wrong,  
  
        የተዘረዘሩትን በየተራ ይምረጧቸውና #fred ይቀጥልን #d ይጫኑ  
  
        ከዚያም #fred ወደኋላ #d የሚለውን ቁልፍ በመጠቀም ሌሎቹን ይመልከቱ')  
else text ('#e በጣም ጥሩ ነው ይቀጥሉበት').  
    button ('ይቀጥል','go_on',7,23).  
end.
```

topic 'go_on'.

```
count = 0.  
move_window (?wq,10,4).  
hide_window (?wq).  
show_window (?w1).  
end.
```

The source code for a sample test (TESTS.KB) is given below.

```

res_count is 0.
topic 'test_quest'.
  text ('#e #x1 #y1 1. አንድ የኮምፒዩተር ሲስተም የሚከተሉት መሠረታዊ ክፍሎች ሊኖሩት ይገባል').
  tq1 is radio_button ([[ 'ሲፒዩ አና ሲስተም ሶፍትዌር',3,2], [ 'ከርድዌርና ሶፍትዌር',33,2],
    [ 'ሀርድዌር ብቻ',3,3,2], [ 'የግባትና የውጣት ኪኖች',33,3,2]]).
  text ('#x1 #y5 2. ከሚከተሉት የቆጠራ ስርዓቶች ውስጥ በኮምፒዩተር ይታዩ የማቀነባበሪያ ክንዋኔ
    ውስጥ አስፈላጊ የሆነው የትኛው ነው-#ren ?#rd').
  tq2 is radio_button ([[ 'የዲስኮ ማሳየት የቆጠራ ስርዓት',3,7], [ 'የባይነት የቆጠራ ስርዓት',35,7],
    [ 'ሌክሰ ጻሊ ማሳየት የቆጠራ ስርዓት',3,8,2], [ 'መልሱ የለም',35,8,2]]).

  text ('#x1 #y10 3. ትንሹ የዳታ መለኪያ የትኛው ነው-#ren ?#rd').
  tq3 is radio_button ([[ 'ቢት',3,11], [ 'ባይት'], [ 'ዲስኮ ማሳየት ቁጥር'],
    [ 'ኪሎባይት']]).

  text ('#x1 #y13 4. የ640 ኪሎ ባይት አካል የትኛው ነው-#ren ?#rd').
  tq4 is radio_button ([[ '640,000 ቢቶች',3,14], [ '640,000 ባይቶች'], [ '650 ባይቶች'],
    [ 'መልሱ የለም']]).

  text ('#x1 #y16 5. ከሚከተሉት አንዱ የማህደረ ትውስታ ዓይነት አይደለም').
  tq5 is radio_button ([[ 'ራም',3,17], [ 'ማግኔታዊ ቴፕ'], [ 'ማግኔታዊ ሸክላ'], [ 'ማግኔታዊ ቅጂ']]).

nextscr ().

if radio_value (?tq1) is 'ከርድዌርና ሶፍትዌር'
  then res_count = ?res_count + 1
  else res_count = ?res_count.

if radio_value (?tq2) is 'የባይነት የቆጠራ ስርዓት'
  then res_count = ?res_count + 1
  else res_count = ?res_count.

if radio_value (?tq3) is 'ቢት'
  then res_count = ?res_count + 1
  else res_count = ?res_count.

if radio_value (?tq4) is '640,000 ባይቶች'
  then res_count = ?res_count + 1
  else res_count = ?res_count.

if radio_value (?tq5) is 'ማግኔታዊ ቅጂ'
  then res_count = ?res_count + 1
  else res_count = ?res_count.

text ('#e #x1 #y1 6. መረጃን ወደ ኮምፒዩተር ውስጥ ለማስገባት የሚያገለግለው ኪን ሞን በመባል
  ይታወቃል#ren ?#rd').
tq6 is radio_button ([[ 'ስክሪን',3,2], [ 'ማተሚያ'], [ 'ሲፒዩ'],
  [ 'መልሱ የለም']]).

```

text ('#x1 #y4 7. ከሚከተሉት ውስጥ ስለሸክላዎች አውነት የሆነው የትኛው ነው#ren ?#rd').

tq7 is radio_button ([['5.25 ኢ.ንች አቅመትን ሸክላ 360,000 ቢቶች ይይዛል', 3, 5],
['3.5 ኢ.ንች አቅመትን ሸክላ 720,000 ቢቶች ይይዛል', 3, 6.2],
['5.25 ኢ.ንች አቅመት ለቅ ሸክላ 720,000 ቢቶች ይይዛል', 3, 7.4],
['3.5 ኢ.ንች አቅመት ለቅ ሸክላ 360,000 ቢቶች ይይዛል', 3, 8.6]]).

text ('#x1 #y11 8. ከምጥርዮተሩን አንድን ስራ አንዲያከናውን የሚነግር የትዕዛዞች ዝርዝር ምን ይባላል#ren ?#rd').

tq8 is radio_button ([['ሲፒዩ', 3, 12], ['ፕሮግራም', ['ቋንቋ', ['አጋፋፊ']]]).

text ('#x1 #y14 9. ከሚከተሉት አንዱ የአፕሊኬሽን ሶፍትዌር ምሳሌ አይደለም').

tq9 is radio_button ([['ወርድፕሮሌሎክት', 3, 15], ['ዲቤዥ', ['ሎተስ 1.2.3'], ['ኤም. ኤስ. ዶስ']]]).

text ('#x1 #y17 10. ፅሁፍን ለመፍጠር፣ ለማቀናጀት፣ አቀማመጡን ለማስተካከል፣ ለማስቀመጥ፣ ለማተምና የመሳሰሉትን ለማከናወን የሚያገለግል የአፕሊኬሽን ሶፍትዌር አይነት የትኛው ነው#ren ?#rd').

tq10 is radio_button ([['ፅሁፍ ማቀናጀት', 3, 19], ['ሆንጠረዥ ትንተና', ['መረጃ አጠናቃፊ', ['መልሱ የለም']]]).

nextscr ().

if radio_value (?tq6) is 'መልሱ የለም'
then res_count = ?res_count + 1
else res_count = ?res_count.

if radio_value (?tq7) is '3.5 ኢ.ንች አቅመትን ሸክላ 720,000 ቢቶች ይይዛል'
then res_count = ?res_count + 1
else res_count = ?res_count.

if radio_value (?tq8) is 'ፕሮግራም'
then res_count = ?res_count + 1
else res_count = ?res_count.

if radio_value (?tq9) is 'ኤም. ኤስ. ዶስ'
then res_count = ?res_count + 1
else res_count = ?res_count.

if radio_value (?tq10) is 'ፅሁፍ ማቀናጀት'
then res_count = ?res_count + 1
else res_count = ?res_count.

text ('#e #x1 #y1 11. ከምጥርዮተር በአንድ ጊዜ የሚያቀናጁት ቢቶች ምን በመባል ይታወቃሉ#ren ?#rd').

tq11 is radio_button ([['ባይት', 3, 2], ['ወርድ', ['ራም', ['ሸክላ']]]).

text ('#x1 #y4 12. የባይነፊ የቆጠራ ስርዓት ስንት መሠረታዊ ምልክቶች አሉት#ren ?#rd').

tq12 is radio_button ([['አስር', 3, 5], ['ሁለት', ['ስምንት', ['አስራ ስድስት']]]).

text ('#x1 #y7 13. የ100011 ባይነፊ አኩያ በዲሲማል ምን ያህል ነው#ren ?#rd').

tq13 is radio_button ([['16', 3, 8], ['19', ['35', ['67']]]).

```

text ('#x1 #y10 14. ከሚከተሉት ውስጥ መረጃን የማቀናጃ መመዘኛ ያልሆነው የተኛው
ካው-#ren ?#rd').
tq14 is radio_button ([[ 'ትክክለኛነት', 3, 11], [ 'ወቅታዊነት', [ 'ፍቺያማነት', [ 'መልሱ የለም' ] ] ]).
text ('#x1 #y13 15. በተወሰነ አካባቢ ከምግብተኞችን የሚያገናኝ መረብ ምን በመባል
ይታወቃል#ren ?#rd').
tq15 is radio_button ([[ 'ሞገ', 3, 14], [ 'ላጎ', [ 'ሞደም', [ 'ራም' ] ] ] ]).

button ('ይታረም', continue, 7, 23).
wait ().

```

```

if radio_value (?tq11) is 'ወርድ'
  then res_count = ?res_count + 1
  else res_count = ?res_count.

```

```

if radio_value (?tq12) is 'ሁለት'
  then res_count = ?res_count + 1
  else res_count = ?res_count.

```

```

if radio_value (?tq13) is '35'
  then res_count = ?res_count + 1
  else res_count = ?res_count.

```

```

if radio_value (?tq14) is 'መልሱ የለም'
  then res_count = ?res_count + 1
  else res_count = ?res_count.

```

```

if radio_value (?tq15) is 'ላጎ'
  then res_count = ?res_count + 1
  else res_count = ?res_count.

```

```
end. (*test_quest*)
```

```

topic feedback.
  if ?res_count < 10
    then revise ()
    else pass ().
end.

```

```

topic revise.
ave = format_number ('###.##%', (?res_count/15) * 100).
text ('#e #x10 የፈተና ውጤትዎ #v ?ave. #v ካው፤ ማለፊያውም 75% ነበር:::

#fred በቂ ውጤት ስላላገኘ
ትምህርቱን ባጠቃላይ ቢደግሙት ይመረጣል #d').

```

```

button ('ይቀጥል', 'go_on', 7, 23).
wait (, 30).
res_count is 0.
end.

```

topic pass.

ave = format_number ('###.##',(?res_count/15) * 100).

text ('#e #x10 የፈተና ውጤትዎ #v ?ave. #v ጾ ነው፤ ማለፊያውም 75% ነበር:::

በመቀጠልም አንደ ዝንባሌዎና አንደ ሥራዎ ጠባይ የሚከተሉትን ኮርሶች ያጥኑ

1. ወርደፕሮግራም አገልግሎት ወይም ወርደ ፅሁፍ ለማቀናጀት
2. ሎተስ 1-2-3 ወይም ኤክሴል በቁጥር ላይ የተመሰረተ መረጃዎችን ለመተንተን
3. ዲቤዥ ወይም አክሶስ መረጃን ለማጠናቀር

#rb ለቀሪዎቹ ኮርሶች መልካም ዕድል አንመኝልዎታለን #rd ').

res_count is 0.

button ('ይቀጥል','go_on',7,23).

wait (,30).

end.

topic radio_value (handles).

selected is element (?handles,

where (get_radio_button (?handles), t)).

radio_value is get_text (?selected).

end.

topic nextscr.

text ('#x1 #y21 #fred ፈተናው ይቀጥላል::: አምስቱን ጥያቄ በአርግጠኝነት ከሰሩ በኋላ ይታረምን ሲመርጡ ሌላ አምስት ጥያቄ ይቀርብልዎታል::: #d').

button ('ይታረም',continue,7,23).

wait ().

end.

topic pass.

ave = format_number ('###.##',(?res_count/15) * 100).

text ('#e #x10 የፈተና ውጤትዎ #v ?ave. #v % ነው፤ ማለፊያውም 75% ነበር።

በመቀጠልም አንደ ዝንግሌዎና አንደ ሥራዎ ጠባይ የሚከተሉትን ኮርሶች ያጥኑ

1. ወርደፕሮግራም ወይም ወርደ ፊልት ለማቀናጀት
2. ሎተስ 1-2-3 ወይም ኤክሴል በቁጥር ላይ የተመረከቱ መረጃዎችን ለመተንተን
3. ዲቤዥ ወይም አክሶስ መረጃን ለማጠናቀር

#rb ለቀሪዎቹ ኮርሶች መልካም ዕድል አንመኝልዎታለን #rd).

res_count is 0.

button ('ይቀጥል','go_on',7,23).

wait (,30).

end.

topic radio_value (handles).

selected is element (?handles,

where (get_radio_button (?handles), t)).

radio_value is get_text (?selected).

end.

topic nextscr.

text (#x1 #y21 #fred ፈተናው ይቀጥላል።። አምስቱን ጥያቄ በአርግጠኝነት ከሰሩ በኋላ ይታረምን ሲመርጡ ሌላ አምስት ጥያቄ ይቀርብልዎታል።። #d').


button ('ይታረም',continue,7,23).

wait ().

end.

DECLARATION

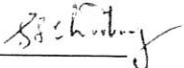
The thesis is my original work and has not been presented for a degree in any other university.



(Signed)

Girma Dessalegn Mekonnen May 17, 1996

The thesis has been submitted for examination with our approval as a university advisors.



(Signed)

Dr. G. G. Chowdhury

May 17, 1996

(Signed)

Ato Nega Alemayehu

May 17, 1996