

**Faculty of Computer Science & Information Technology**  
**University Of Malaya**  
**50603 Kuala Lumpur**  
**Malaysia**

**Project Title** Multimedia Teaching Package  
-Teaching XML

**Supervisor** Prof. Madya Dr. P.Sellapan

**Moderator** Encik Chiew Thiam Kian

**Name** Semeren Kaur A/P Jaswanth Singh

**Matrix No** WEK 990141

**Dissertation submitted by in fulfillment of the requirements for the  
Degree of Bachelor of Computer Science**

## ABSTRACT

Multimedia Teaching Package is software that teaches users on a particular subject or matter. It contains multimedia elements such as text, graphics, images, animation, video and sound that are integrated to create an interactive teaching tool. . Multimedia development teams have long been challenged by educationalists to provide sophisticated learning environments that empower the learner to use the computer as a cognitive tool that overcomes the limitations of existing environments. Cognitive tools enable learners to actively construct and transform their understanding of an area of learning, test their developing ideas and reflect on the products of their learning.

The system that I am developing is a Multimedia Teaching Package that teaches XML. This software is called Learning XML. This system is a tool where people of different age levels and professional skills can use at any time . One of its objective is to so provide an easy learning tool to the user compared to textbooks. This is what I would like to achieve from creating the teaching package. Users who are interested in learning a language get discouraged after looking at a 300 page book that can help them learn something. With this software I m sure more users will be keen on learning XML , a new hypertext markup language which is often compared to HTML and SGML.

There are many developing tools that were researched and after going through a through look over the tools and the tools that were essential were chosen. They are explained in Chapter 3 .Basically it covers Authoring Tools such as Macromedia Director, Macromedia Flash , Adobe Photoshop and many more software that will enhance the betterment of my system.

Methodology is a process that is used to develop a system. It is important to understand the nature of a program when developing a system. There are many methodologies that can be used such as the Waterfall Model, the V-Shape Model, Spiral Model methodology .The methodology adopted for this project is a modified version combining the benefits of both waterfall model and prototyping approach . Prototyping can be viewed as a risk reduction technique. Experiments have shown that prototyping reduces the number of problems connected to requirement



specifications and the overall development cost may be lower if a prototype is developed. Requirements for the project was gathered such as the Functional Requirement , Non-Functional Requirement , Prototyping Requirement and other requirements to enhance the system so a better package can be delivered.

Finally in the System Design ,an overview of the chapters were done and what that is going to thought in the system were introduced . The DFD's for this system were also introduced . A DFD is a graphical illustration that shows the flow of data and logic within a system. The interface of the system was created to show on how the interfaces will look when the system will be created.

Now that I have completed the system...yes Learning XML is ready to be evaluated by users. In this 3 months there have been many changes in the system .This is due to some problem that couldn't be avoided and due to time constraint .Hopefully the system has met the user requirements.

In chapter 5 System Development and Implementation is where most of the time required. System Development was a critical stage when I was developing the teaching package and it has been the most important stage because this is where your research on the 4 earlier chapters have to be implemented.

In System Testing, the issues covered would be regarding testing. It starts of by a short and simple explanation about what is testing. It then describes the testing strategy/plan used in this project. Next, it moves on to which type of testing would be applied in this project and why they are used. It also includes the various test plans, which will be drawn up when testing is implemented.

And finally the system evaluation and Conclusion was done. Hope all the evaluation, problems and will be solved for future reference.

Basically while preparing this documentation , I have learned many things. Some of the software's were new to me and so is the language XML. But this has not kept me a way to create a good system.

## ACKNOWLEDGEMENT

I would like to take this opportunity to express my highest gratitude to my supervisor, Professor Madya Dr. P Sellapan for his valuable advice, guidance , encouragement , and support along this project.

I would also like to thank Mr. Chiew Thiam Kian , my moderator for being such a generous person and taking his time out to guide students like me and have been very understanding towards some of the problems faced towards completing the system.

I would like to express my warmest appreciation to my course mates, Fislin, who in a short period has become a good friend ,Serena Kaur, Mary Sivam, How Men Mee, LourduMary and my other fellow course mates whom have given me the support to finish this project. And not forgetting some of my seniors who have guided me in this project.

Last but not least I would like to express my deepest gratitude to my family whom have supported me. They have encouraged me and always stood by me to make sure that I didn't give up.

Thank You,

Semeran Kaur A/P Jaswanth Singh

WEK 990141



**LIST OF FIGURES****Chapter 1 : Introduction**

Figure 1.0	Project Schedule	9
------------	------------------	---

**Chapter 2-Literature Review**

Figure 2.0	The Authoring Process	36
------------	-----------------------	----

**Chapter 3- Methodology**

Figure 3.1	System Development Model adopted for this project	57
------------	---	----

**Chapter 4 – System Design**

Figure 4.1	Representing the structured module of Learning XML	78
Figure 4.2	Context Diagram For Learning XML	79
Figure 4.3	Level Zero of Data Flow Diagram	80
Figure 4.4	Level One of the DFD for Selection Process	81
Figure 4.5	Basic Database Design	82
Figure 4.6	Figure 4.6 The example of the main interface	84
Figure 4.7	The interface on Lessons	85
Figure 4.8	The interface for the selection on tutorials.	86

**Chapter 5 System Development and Implementation**

Figure 5.1	The main page –an interface in Macromedia Director 8.0	91
Figure 5.2	The Interface of The Score of The Window	92
Figure 5.3	The Interface of The Main Screen in Learning XML	94

## Table of Contents

**Chapter 6 System Testing**

Figure 6.1 Verification and Validation Of Static and Dynamic	106
Figure 6.2 Types Of Testing Identified in This Project	107
Figure 6.3 Components of System Testing	110

## Chapter 1-Introduction

## 1.1 Introduction

## 1.2 Project Overview

## 1.3 Objectives

## 1.4 Project Scope

## 1.5 Project Significance

## 1.6 Project Outcome

## 1.7 Report Layout

## 1.8 Project Schedule

## Chapter Summary

## List Of References

## Chapter 2-Literature Review

## 2.1 Purpose of Literature Review

## 2.2 Project Background Study



**Table Of Contents**

2.2.1 An Introduction on XML	12
<b>Abstract</b>	
<b>Acknowledgement</b>	10
<b>List of Figures</b>	
2.3.1 Where can be used	16
<b>Chapter 1-Introduction</b>	
1.1 Introduction	1
1.2 Project Overview	2
1.3 Objectives	3
1.4 Project Scope	5
1.5 Project Significance	6
1.6 Project Outcome	6
1.7 Report Layout	7
1.8 Project Schedule	9
Chapter Summary	10
List Of Reference	11
2.7.3 Power Point	32
2.7.4 Zoomed Room	33
<b>Chapter 2-Literature Review</b>	
2.8 Development Tools	
2.1 Purpose of Literature Review	12
2.2 Project Background Study	12

2.2.1	An Introduction on XML	12
2.3	Introduction To Multimedia	16
2.3.1	Where can be used	16
2.4	Interactive Multimedia	17
2.4.1	Why Multimedia is used	20
2.5	Types of Media included to create a multimedia production	22
2.6	The components of Interactive Multimedia	25
2.7	Findings	28
2.7.1	Survey And Questionnaire	29
2.7.2	Using the Internet	30
2.7.3	Research on Books	32
2.7.4	Document Room	33
2.8	Development Tools	
2.8.1	What is Authoring Tool	34
2.8.2	Authoring Process	35



2.8.3 Introduction on Authoring Tools	36
2.8.4 Scripting For Multimedia	43
2.8.5 Other Tools	45
2.8.6 Comparison on Authoring Tools	47
2.9 Comparison of Existing System	
2.9.1 TECFA's XML page	49
2.9.2 Working with XML	50
2.9.3 XML Teaching	51
2.10 Project Outcome	51
Chapter Summary	51
List of References	52
<b>Chapter 3- Methodology</b>	
3.1 An Introduction on System Analysis	53
3.2 Overview Of the Methodology Process	54
3.3 System Development Methodology	55

---

3.3.1	Justification	58
3.3.2	Strength	58
3.3.3	Weakness	59
3.4	Software Development Tools Analysis	
3.4.1	Authoring Tools	59
3.4.2	Programming Tools	62
3.4.3	Other Tools	63
3.5	Requirement Analysis	65
3.5.1	Requirement Elicitation	65
3.5.2	Functional Requirements	66
3.5.3	Non Functional Requirement	67
3.5.4	Prototyping Requirements	68
3.5.5	Runtime Requirements	68
	Chapter Summary	70
	List Of References	71

---



List Of Reference	87
-------------------	----

## **CHAPTER 5- SYSTEM DEVELOPMENT AND IMPLEMENTATION**

5.1 Introduction	88
5.2 Development Environment	88
5.2.1 Hardware Tool	88
5.2.2 The software required during testing	88
5.3 Development Of the System	89
5.3.1 A Brief Introduction About Director 8.0	89
5.3.2 The Development of Modules	92
5.3.3. The Development of the Screen Interfaces	94
5.3.4 Buttons and Icons	95
5.4 Coding	96
5.4.1 Extensible Markup Language (XML)	96
5.4.2 Lingo Scripting	96
5.4.3 Structure Programming	101
5.5. System Documentation	102

Summary	103
List Of References	104
<b>CHAPTER 6-SYSTEM TESTING</b>	<b>105</b>
6.1 Introduction	105
6.2 Testing Objectives	105
6.2.1 What is Testing	105
6.3 Types of Testing Identified in This Project	107
6.4 Types of Testing Techniques Used	108
6.4.1 Unit Testing	108
6.4.2 Integration Testing	108
6.4.3 Validation Testing	108
6.4.4 System Testing	109
6.5 Testing Personnel	110
6.5.1 University Malaya Students	110
6.6 Testing Environment	111
6.7 Testing Hardware	111

6.8 Testing Software	111
Summary	112
List Of References	113

## **CHAPTER 7-SYSTEM EVALUATION AND CONCLUSION**

7.1 Introduction	114
7.2 Problems and Solutions	114
7.3 Features And Strength	119
7.4 System Limitations	120
7.5 Future Enhancement	121
7.6 Gained Knowledge and Experience	122
Summary	123

## **USER MANUAL**

## **APPENDIX**



## Chapter 1: INTRODUCTION

### 1.1 Introduction

As we are now in the beginning of the new millennium, we are being exposed to many new technologies that are having a profound effect on our daily life. Technology have been introduced to our lives in many ways.

Multimedia Teaching Package is software that teaches concepts on a particular subject or matter. It contains multimedia elements such as text, graphics, images, animation, video and sound that are integrated to create an interactive learning environment.

# CHAPTER I INTRODUCTION

A flexible computer-based learning package is being developed as part of an effort to enhance the quality of education. This package, which will contain CD-ROM, has been developed to meet the needs of the new generation of students who are getting popular since the 21st century. The multimedia teaching package contains interactive multimedia elements such as text, graphics, images, animation, video and sound that are integrated to create an interactive learning environment. This material together with a detailed text and graphics manual, can now be made available to users who can review the material in their own time, explore various parts of the material, and interactively review information and information. Users can use the package individually and in small groups.

## Chapter 1 : INTRODUCTION

### 1.1 Introduction

As we are now in the beginning of the new millennium, we are being exposed to many new technologies that are having a profound effect on our daily life. Technology have been introduce in our lives in many ways.

Multimedia Teaching Package is software that teaches users on a particular subject or matter. It contains multimedia elements such as text, graphics, images, animation, video and sound that are integrated to create an interactive teaching tool. . Multimedia development teams have long been challenged by educationalists to provide sophisticated learning environments that empower the learner to use the computer as a cognitive tool (Rowe, [1]) that overcomes the limitations of existing environments. Cognitive tools enable learners to actively construct and transform their understanding of an area of learning, test their developing ideas and reflect on the products of their learning.

A flexible computer-based learning package is being developed so that can be used to enhance the users ability to learn XML. The package, which will available on CDROM, has been designed to teach users the new hypertext markup language that is getting popular these days – XML. The multimedia teaching package contains resource materials that can be used in a large theatre, including video clips, sound recordings and overhead projection slides. This material, together with additional text and graphic material, can then be made available to users who can revise the material in their own time, explore aspects of it in more depth, and interactively access references and information. Users can use the package individually and in small groups.



## 1.2 Project Overview

The multimedia teaching package that is going to be developed is called Learning XML. Learning XML is a teaching package that will help users learn XML (Extensible Markup Language). XML [2] is pretty simple, and very flexible. It has many uses yet to be discovered – developers are just beginning to scratch the surface of its potential. It is the foundation for a great many standards yet to come, providing a common language that different computer systems can use to exchange data with one another. As each industry-group comes up with standards for what they want to say, computers will begin to link to each other in ways previously unimaginable.

The teaching package will help users learn XML in an interactive way. It contains multimedia elements that will enhance the users ability to learn XML. People from different background and age level can use it. The programs clear, patient and structured delivery will help users to understand and will assist them in using the software better. This will help the users in developing confident in using this teaching package.

Now something on multimedia, multimedia is a multiple means by which information is stored, transmitted, presented or perceived. It contains five important elements that are text, graphics, animation, video and sound that are integrated to create a production. More information on multimedia and why is it used in this teaching package will be explained in chapter 2.

The main target users of this program are software engineers who develop software's in their respective companies. This will be useful to those who are dealing with language Java and HTML. Many companies deal with creating software using Java and uses HTML if they are working on a web-based program. Since XML is becoming popular and its usage can be expanded to many different aspects, more people are using it. So this will make it easier for these people who are interested in learning XML.



Students who are taking computer science in their universities and colleges can also use it. They can use it to help them understand XML better. So can the lecturers of these higher institutions. They can use this teaching package to teach the students in the lecture halls. This can make their classes more interesting. Traditional approaches to tertiary teaching are based on a transmission model of teaching where the teacher delivers their knowledge to a large group of students. This approach is very robust because it is easy and efficient and requires no professional education training. Educational innovations, grounded in Dewey's theory of inquiry learning [3], continue to change learning environments so that the role of the teacher changes from the 'teller' to a 'facilitator of learning'. With using this teaching package, the normal routine can be changed and students can learn in more interactive way.

This multimedia teaching package is divided into six main modules. All this modules provides an interesting learning environment with integration of text, graphics, animations, video and sound. It computerizes the stiff book into an interactive learning package and it's a step towards the MSC objectives. It's a much better tool; more efficient is sending the information across to users compared to a book.

### 1.3 Objectives

As mentioned before there aren't any XML Teaching Packages currently available in market. Those who are interested to learn XML find it difficult in getting a comprehensive tool in teaching XML. With this, Learning XML has been developed with the following objectives:

- 1) To provide an easy learning tool to the user compared to textbooks.

This teaching package provides step by step learning guides for users. Each and every instruction clearly defines on what is going to happen next and what is happening. It has appendix and notes to help users find what they need without going through the whole package.

- 2) To provide a multimedia interactive and an interesting learning environment.

With an interactive user interface and the integration of the multimedia elements, attracts the users interest. In this way, it makes the learning process easier and the users would be eager to learn XML.

- 3) The programs clear, patient and structured delivery will create a better understanding on XML.

The structure and the layout of the package is very clear where everything is explained and it's easy to use. Users will not get confused with the layout and the flow of delivery will be easy to understand.

- 4) To computerize the stiff textbook into an interactive learning package to build users interest in learning XML.

This is to gather all the information that have to presented in at interactive way that users wont get bored reading it and using the software. As a matter of fact they will be more interested in using the package than a textbook which only have pictures and text which does not create a stimulation environment to learn.

- 5) To develop a learning package that reach people at different learning styles and skill level.

This package will be able to reach people at any age level with different skill levels. It can be used by professionals or students at any age level.

- 6) To improve the feature of computer software on teaching XML available in market.

There are not many XML teaching software now in the market that provides guidance to users. This will be another teaching tool that users can use to learn better.



## 1.4 Project Scope

A Window based application which will teach users the basic XML structure.

Consist of 6 modules :

- **Introduction**-An introduction on XML. It contains the history and the main usage of XML. It explains why XML is becoming more popular and what advantages it has compared to SGML and HTML.
- **Lessons** - It is divided into 8 topics. Each topic has its own contents that will explain and teach users on the language XML. It starts with teaching the user the basic such as elements, tags and attributes, processing instructions, validations on schemas, document type definitions(DTD), writing a file , designing a document and XML with other applications such as Java, XSL and other applications.
- **Tutorial** – It contains questions concerning the lessons.
- **Library**- A repository on XML coding and other important facts. This is to help users to get ready codes and tags so it will be easier for them to use the create their own XML documents.
- **Appendix**- The important keywords in XML and the teaching package. It contains all the important keywords in the package so users can Find,the information needed without having to navigate through the whole package.



## 1.5 Project Significance

- It promotes active learning rather than passive learning
- It motivates users from different backgrounds.
- It enables users to learn at their own pace.
- It allows users choices of which topics and aspects they will cover each time they use the software.
- It caters to those people who are more visually oriented and who are often disadvantaged by a more textual or verbal approach.
- It utilizes a variety of formats such as sound, video, animation, text and graphics in one package to maintain interest and attention.
- It promotes critical thinking and analysis of complex issues and programs.

## 1.6 Project Outcome

A major outcome of this project is the development of the multimedia teaching package. The package fits onto a CDROM, which can be used at any time and used by anyone with different styles and different skills. The package incorporates a variety of resource materials including text, graphics, video clips, sound clips and animations. It provides the information for users to a link to the newsgroups on the Internet and references to library material. It also includes a way for students to test their own knowledge in a non-threatening and non-assessable manner. Users are able to search the database by key words, or use various indexes to find any of the content of the package.

This multi-media package contains a variety of materials including video clips, sound recordings and color layout that can be easily accessed and presented by the users using projection directly from the computer to a large screen. This to add variety and interest to the presentation so that can stimulate interactive activities in a large theatre. These activities improve the learning process by keeping the length of concentrated listening to short manageable periods and by promoting active learning and wider participation by users.

## 1.7 Report Layout

The purpose of this layout is to give an overview of the major phases involved in the development of the project. Below is the report layout :

### Chapter 1 : Introduction

This chapter gives an overview of projects objectives, project's scope , significance and its outcome. It also shows the project schedule on how much time is taken to complete the project.

### Chapter 2 :Literature Review

This chapter covers the project background study that is an introduction to XML and the differences between HTML and SGML. It also explains about the terms, the elements and the importance of multimedia and its usage in the package. This chapter also has findings of the project on how information and requirements were gathered. The introduction on authoring tools and the comparison for these tools were done in this chapter. The comparison on the existing system is also explained in this chapter.

### Chapter 3 : Methodology

This chapter includes the system analysis that was done. It also contains the methodology that is used in creating this multimedia teaching package. Here is where the chosen tools are explained.



All the requirements of the system is including the non-functional, functional and prototyping requirements are explored in this chapter.

#### **Chapter 4 : System Design**

This chapter describes the logical design of this system. It includes the systems modules design, system data flow design and user interface design. It explains about the Human –Computer Interaction and the Graphical User Interface.

#### **Chapter 5 : System Development and Implementation**

This section will discuss the development strategy as well as type of implementations for the whole package. System Development is one of the critical stages in developing a system. System implementation is a process that converts the system requirements and design into programming codes. This phase at some point involves some modifications to the previous design.

#### **Chapter 6: System Testing**

In this section , the issues covered would be regarding testing. It starts of by a short and simple explanation about what is testing. It then describes the testing strategy/plan used in this project. Next, it moves on to which type of testing would be applied in this project and why they are used. It also includes the various test plans, which will be drawn up when testing is implemented

#### **Chapter 7: System Evaluation And Conclusion**

This chapter looks into one of the important topics regarding this project report. It highlights issues such as the success of the project in academic terms. It looks into the degree of success in carrying out the project; learning experienced gained, things to be done differently and the value or the learning process and how it contributes to my professional and academic expertise.



## 1.8 Project Schedule

In this chapter the introduction on the multimedia teaching package was done. The teaching package that is going to be developed is called Learning XML. In this chapter the objectives, the project scope and the project significance were laid out to guide me through creating this package. The project outcome is what I will be trying to achieve so that it can be a satisfactory to the users. The project schedule is to adjust what has to be done in the time given to finish the project. Basically this chapter is to help me start the project from the scratch and giving me a outline on what has to be done.

## Chapter Summary

In this chapter the introduction on the multimedia teaching package was done. The teaching package that is going to be developed is called Learning XML. In this chapter the objectives, the project scope and the project significance were laid out to guide me through creating this package. The project outcome is what I will be trying to achieve so that it can be a satisfactory to the users. The project schedule is to adjust what has to be done in the time given to finish the project. Basically this chapter is to help me start the project from the scratch and giving me a outline on what has to be done.

**List Of Reference**

- [1] Rowe, H. (1993) Learning with Personal Computers ,Hawhorn ACER
- [2] Simpson, John E.,Just XML,2<sup>nd</sup> Edition,Prentice Hall, 2001
- [3] Dewey, J(1990) How We Think, Boston : Health



## CHAPTER 2: LITERATURE REVIEW

### 2.1 Purpose Of Literature Review

The review of literature describes the findings of various researchers and technology for the effectiveness of the multimedia teaching package. The idea, knowledge and experience gained during the research regarding the project overview, the development, tests and others will be used in the development of the multimedia teaching package.

Various good and relevant features are to be used during the survey, particularly the design and interface created already by various systems. Thus, the literature review provides pertinent information and history to the research and the researcher will be necessary to determine and understand the best solution.

### 2.2 Project Description

#### 2.2.1

Structure

of value

learning from contact in a document, which is something different than contact in a

figure representing contact in a document, which is something different than contact in a

structure. A model of learning is a document, which is something different than contact in a

XML specification defines a document, which is something different than contact in a

#### What's a Document?

The number of applications currently being developed that are based on, or make use of, XML documents is truly amazing. Due to considering that XML is very new. For our purposes, the word "document" refers not only to traditional documents, like this one, but also to the kind of other XML "documents" that include vector graphics, e-commerce transactions, multimedia of a picture, object-oriented data, voice APIs, and a thousand other kinds of structured information.

## CHAPTER 2 : LITERATURE REVIEW

### 2.1 Purpose Of Literature Review

The review of literature describes the findings of various researches and technology for the effectiveness of the multimedia teaching package. The idea, knowledge and experience gained during the research regarding the project overview, the development tools and others will be used in the development of the multimedia teaching package. Various good and relevant features are to be noted during the survey, particularly the design and interfaces method used by various systems. Thus, the literature review provides pertinent information and validity to the research and the environment will be necessary to determine and implement the best solution.

### 2.2 Project Background Study

#### 2.2.1 An Introduction On XML

Structured information contains both content (words, pictures, etc.) and some indication of what role that content plays (for example, content in a section heading has a different meaning from content in a footnote, which means something different than content in a figure caption or content in a database table, etc.). Almost all documents have some structure. A markup language is a mechanism to identify structures in a document. The XML specification defines a standard way to add markup to documents.

#### What's a Document?

The number of applications currently being developed that are based on, or make use of, XML documents is truly amazing particularly considering that XML is very new. For our purposes, the word "document" refers not only to traditional documents, like this one, but also to the myriad of other XML "data formats". These include vector graphics, e-commerce transactions, mathematical equations, object meta-data, server APIs, and a thousand other kinds of structured information.



### Is XML Just Like HTML?

No. In HTML, both the tag semantics and the tag set are fixed. An `<h1>` is always a first level heading and the tag `<ati.product.code>` is meaningless. The W3C, in conjunction with browser vendors and the WWW community, is constantly working to extend the definition of HTML to allow new tags to keep pace with changing technology and to bring variations in presentation (style sheets) to the Web. However, these changes are always rigidly confined by what the browser vendors have implemented and by the fact that backward compatibility is paramount. And for people who want to disseminate information widely, features supported by only the latest releases of Netscape and Internet Explorer are not useful.

XML specifies neither semantics nor a tag set. In fact XML is really a meta-language for describing markup languages. In other words, XML provides a facility to define tags and the structural relationships between them. Since there's no predefined tag set, there can't be any preconceived semantics. All of the semantics of an XML document will either be defined by the applications that process them or by style sheets.

### Is XML Just Like SGML?

No. Well, yes, sort of. XML is defined as an application profile of SGML. SGML is the Standard Generalized Markup Language defined by ISO 8879. SGML has been the standard, vendor-independent way to maintain repositories of structured documentation for more than a decade, but it is not well suited to serving documents over the web (for a number of technical reasons beyond the scope of this article). Defining XML as an application profile of SGML means that any fully conformant SGML system will be able to read XML documents. However, using and understanding XML documents *does not* require a system that is capable of understanding the full generality of SGML. XML is, roughly speaking, a restricted form of SGML.

For technical purists, it's important to note that there may also be subtle differences between documents as understood by XML systems and those same documents as understood by SGML systems. In particular, treatment of immediately adjacent tags may be different.



## Why XML?

In order to appreciate XML, it is important to understand why it was created. XML was created so that richly structured documents could be used over the web. The only viable alternatives, HTML and SGML, are not practical for this purpose.

HTML, as have discussed, comes bound with a set of semantics and does not provide arbitrary structure.

SGML provides arbitrary structure, but is too difficult to implement just for a web browser. Full SGML systems solve large, complex problems that justify their expense.

Viewing structured documents sent over the web rarely carries such justification.

This is not to say that XML can be expected to completely replace SGML. While XML is being designed to deliver structured content over the web, some of the very features it lacks to make this practical, make SGML a more satisfactory solution for the creation and long-time storage of complex documents. In many organizations, filtering SGML to XML will be the standard procedure for web delivery.

## XML Development Goals

The XML specification sets out the following goals for XML:

1. XML over the Internet. Users must be able to view XML documents as quickly and easily as HTML documents. In practice, this will only be possible when XML browsers are as robust and widely available as HTML browsers, but the principle remains.
2. XML shall support a wide variety of applications. XML should be beneficial to a wide variety of diverse applications: authoring, browsing, content analysis, etc. Although the initial focus is on serving structured documents over the web, it is not meant to narrowly define XML.
3. XML shall be compatible with SGML. Most of the people involved in the XML effort come from organizations that have a large, in some cases staggering, amount of material in SGML. XML was designed pragmatically, to be compatible with existing standards while solving the relatively new problem of sending richly structured documents over the web.



4. It shall be easy to write programs that process XML documents. The colloquial way of expressing this goal while the spec was being developed was that it ought to take about two weeks for a competent computer science graduate student to build a program that can process XML documents.
5. The number of optional features in XML is to be kept to an absolute minimum, ideally zero. Optional features inevitably raise compatibility problems when users want to share documents and sometimes lead to confusion and frustration.
6. XML documents should be human-legible and reasonably clear. If you don't have an XML browser and you've received a hunk of XML from somewhere, you ought to be able to look at it in your favorite text editor and actually figure out what the content means.
7. The XML design should be prepared quickly. Standards efforts are notoriously slow. XML was needed immediately and was developed as quickly as possible.
8. The design of XML shall be formal and concise. In many ways a corollary to rule 4, it essentially means that XML must be expressed in and must be amenable to modern compiler tools and techniques.  
There are a number of technical reasons why the SGML grammar *cannot* be expressed. Writing a proper SGML parser requires handling a variety of rarely used and difficult to parse language features. XML does not.
9. XML documents shall be easy to create. Although there will eventually be sophisticated editors to create and edit XML content, they won't appear immediately. In the interim, it must be possible to create XML documents in other ways: directly in a text editor, with simple shell and Perl scripts, etc.
10. Terseness in XML markup is of minimal importance. Several SGML language features were designed to minimize the amount of typing required to manually key in SGML documents. These features are not supported in XML. From an abstract point of view, these documents are indistinguishable from their more fully specified forms, but supporting these features adds a considerable burden to the SGML parser (or the person writing it, anyway). In addition, most modern editors offer better facilities to define shortcuts when entering text.



## 2.3 Introduction to Multimedia

Multimedia is not a single technology .Multimedia stands for the convergence of several streams of development in the computing, audio, video and communication industries. The term “multimedia” is often used but hard to define. As a buzzword, it is used to advertise different products such as video games, computer –based training, sales presentations etc. Basically, multimedia can be understood as the integration of more than one medium. In this broad interpretation, current media systems of our society such as television and newspapers fall within the scope of multimedia.

Multimedia is a combination of still and moving images, audio, text and interactivity from a pool of human –machine communication media. It could also defined as a software package utilizing high resolution graphics, stereo sound , digitized images ,full motion video, animation and text .The term multimedia has become somewhat a trendy ,hyped word in the present computer age. When distilled to its root elements its meaning can be devised : *multi* (‘much’ or ‘many’) and *media* (from *medium* meaning ‘means of communicating’).Using multimedia then is simply using a variety of media ,whatever visual or auditory ,with the intent of communicating.

Multimedia technology not only makes it possible but relatively easy to extend the power of personal computing far beyond the traditional realm of data processing. As a result , the barriers to expending the role of computers in our daily lives have been erased. Day- to-day applications of multimedia can be visualized by considering activities that resemble the process of communicating information, such as education and training, reference, point-of-information, business presentation, manuals, database and advertising and others.

### 2.3.1 Where can multimedia be used

Multimedia can be used in just any situation where computers are being used to deliver information and a number where they are frequently used .It may just be used to enhance areas that previously contains text only.



For example, presentation of information kiosks. Multimedia e-mail and video conferencing allows information to be communicated better. Collaborative working tools are becoming cheaper and more widely available and video conferencing applications now allow application sharing. Multimedia is widely used in education, from pre-school to postgraduate. Interactive multimedia allows students to explore data and simulations allow the students to work an experiment. Can be particularly useful in a distributed environment with distance learning. Multimedia can be used to enhance data representation and data analysis. It can be done by converting the original form from a satellite to sound, it becomes much easier to hear discrepancies in the data.

## 2.4 Interactive Multimedia

Interactive multimedia has been called a "hybrid technology." It combines the storage and retrieval capabilities of computer database technology with advanced tools for viewing and manipulating these materials. Multimedia has a lot of different connotations, and definitions vary depending on the context.

Interactive multimedia is defined by three criteria:

- Interactive Multimedia is any package of materials that includes some combination of texts, graphics, still images, animation, video, and audio;
- These materials are packaged, integrated, and linked together in some way that offers users the ability to browse, navigate and analyze these materials through various searching and indexing features, as well as the capacity to annotate or personalize these materials;



- Interactive multimedia is always "reader-centered." In interactive multimedia, the reader controls the experience of reading the material by being able to select among multiple choices, choosing unique paths and sequences through the materials. One of the key features of interactive multimedia is the ability to navigate through material in whatever ways are most meaningful for individual users.

Interactive multimedia is synonymous with another frequently used term: hypermedia. Hypermedia is the multimedia version of the term hypertext. A hypertext is defined as any non-sequential, electronic text, assembled not as a seamless sequence of material with a beginning, middle and end, but as a web of interrelated chunks" of text. In a hypertext, the reader controls the sequence of reading by choosing how to navigate among these chunks of text by various electronic links. The term hypermedia" was coined to mean a hypertext that uses multiple media. In other words, hypermedia is a collection of multimedia materials with multiple possible arrangements and sequences. Hypertext and hypermedia are "electronic" concepts that can only exist in a computer-based environment. Only in a computer-based environment can materials can be linked and organized in multiple ways simultaneously, and searched, sorted and navigated in hundreds of possible combinations by different users.

Imagine, for example, a large comprehensive textbook on a programming language such as XML. In a sense, a print version of that textbook is already "multimedia": that is, in addition to text, it might have pictures, maps, graphs, charts, timelines; furthermore, the text is made up of many different texts, being a combination of words written by the author, quotations from other figures, perhaps commentary by other authors and so on.

But while the textbook could be thought of as a text using multiple media and materials, it is not a multimedia hypertext (or hypermedia) because, as a printed book, it can only be arranged in one order; its materials can only be accessed in the one way that the author and the publisher arranged them.



True, a reader can access the print text in non-linear ways by using the index at the back of the book, or by jumping around. Still, the text itself has only one arrangement and one hierarchy of topics; and the reader's ability to navigate the materials is limited by the table of contents and the index.

Furthermore, a printed XML textbook is limited by the constraints of size and practicality. Only so much information can comfortably fit between two covers of a printed book. Such practical issues have important consequences for the kinds of materials that go into the hands of readers. Limitation of size means that it is more practical to write books that synthesize and make reference to large bodies of documents without being able to include very much or any of the documents, themselves--even though, for the author documents are part of the vital material of the book.

Now imagine a XML textbook in electronic form, constructed as a work of hypermedia: how would that work be different from a printed text? First, you could fill that "book" with a far greater number of materials than you could fit between two covers of a printed book (a CD-ROM compact disk, for example, can hold the equivalent of 300,000 pages of printed material). Second, you could have a book that was truly "multimedia" in that, in addition to text, photographs, charts, and timelines, you could have audio (such as folksong recordings, famous speeches), and video (such as newsreels, film clips). Third, an electronic textbook could be constructed in an entirely different way from a printed textbook. It could have dozens of potential organizations, and thousands of internal linkages that could take the reader from one related idea to the next, in ways that would infinitely vary depending on the context of the reading experience and the interests of the reader.

Consequently, the structure of such a "text" would not be limited to the single story line or synthesis offered by the author, but would become an intricate web of interrelationships, something approaching the complexity of the language. In discussing the transformations that ensued in turning the print



version of the book Learning XML into a multimedia CD-ROM, it is noted that the very nature of the "book" changed.

#### 2.4.1. Why use Interactive Multimedia

We've seen that interactive multimedia, by definition, has the capacity to deliver large amounts of materials in multiple forms, and to deliver them in an integrated environment that allows users to control the reading and viewing experience. How then do these defining characteristics and virtues translate into benefits in an educational environment?

First of all, multimedia programs bring to education the extraordinary storage and delivery capabilities of computerized material. This is especially important for schools, libraries, and learning institutions where books are difficult to obtain and update. Multimedia is a powerful and efficient source for acquiring learning resources. Multimedia can also provide educational institutions access to other kinds of inaccessible materials, such as hard to find historical films, rare sound recordings of famous speeches, illustrations from difficult to obtain periodicals, and so on. Multimedia can put primary and secondary source materials at the fingertips of users in even the remotest locations from major research facilities.

Secondly, it is not just sheer access to these materials that makes multimedia a powerful tool, but the control over those materials that it gives to its users. Interactive multimedia programs enable the user to manipulate these materials through a wide variety of powerful linking, sorting, searching and annotating activities.

Each of these activities can be made to reinforce and inculcate various intellectual skills, in addition to satisfying certain cognitive needs for quality learning, such as the ability to follow through links at the immediate



moment when curiosity is aroused, and the ability to view different forms of the same information side-by-side.

Furthermore, interactive multimedia programs usually integrate some combination of orientation tools, such as timelines, graphs, glossaries, and other pedagogical guides. These kinds of tools further point to the third major benefit of multimedia: the personalization or individualization of the learning experience.

By allowing users to control the sequence and the pacing of the materials, multimedia packages facilitate greater individualization in learning, allowing students to proceed at their own pace in a tailored learning environment. Furthermore, interactive multimedia can be a powerful learning and teaching tool because it engages multiple senses. Students using multimedia are reading, seeing, hearing, and actively manipulating materials. As one educator enthusiastically put it,

As humans, we seem hard-wired for multiple input. Consider that we remember only about 10% of what we read; 20%, if we hear it; 30%, if we can see visuals related to what we're hearing; 50%, if we watch someone do something while explaining it; but almost 90%, if we do the job ourselves-- if only as a simulation. In other words, interactive multimedia--properly developed and properly implemented-- could revolutionize education. (Menn, 1993)

Although "revolutionize" may be a bit optimistic, interactive multimedia is a promising medium for reinforcing, extending, and "supplementing" what goes on in the classroom with print materials, lectures and classroom discussions.

I use the term "supplementing" quite intentionally, however, as the supplementary dimension of multimedia materials is important to keep in mind. Incorporating multimedia into the curriculum does not mean "throwing out the printed books." Most teachers who incorporate some



kind of interactive multimedia into their teaching do so to enhance printed materials and the core course content. Multimedia materials help students and teachers by way of reinforcement and extension, not substitutions. What hypermedia provides is access to materials and unique personalized control over them. In other words, interactive multimedia isn't about replacing books, but about replacing the absence of books; hypermedia doesn't do what books do, but what books can't do.

## **2.5 Types of media included to create a multimedia production**

As have been mentioned earlier that multimedia is a multiple means by which information is stored, transmitted, presented or perceived. The basic elements in an interactive multimedia include text, graphics, animation, audio and video. Human-Computer Interaction is also a vital ingredient in interactive multimedia.

### **Text**

Words and symbols in any form, spoken or written is the most common system of communications. They deliver the most widely understood meaning of the greatest number of people-accurately and detail. Because of this, they are vital elements of multimedia menus, navigation systems and content. Text acts as the keystone tying all the other media elements together whereas a picture is only worth a thousand words. With multimedia technology, text can be combination with other media in a powerful and meaningful way to present information and to create a production.

### **Graphics**

The introduction of Apple Macintosh computer and the Microsoft Windows program changed the way we worked with computer. Using a mouse and a desktop metaphor, we click on icons and drop-down menus, drag folders and resized windows. We are accustomed to working with graphical images on the screen and in fact expect to see them. Graphics are used to :

- Add color and images to the page background
- Enhance spatial design to the page layout
- Communicate personal styles
- Display graphic artwork and photographs
- Display spreadsheets and graphs
- Create text graphics for layout effects

### **Sound**

Sound is a potent communication medium. We can use sound effects ,music and narration to enhance the multimedia application. Sound files such as \*.au , \*.wav and \*.aif formats tend to be large depending on the quality and supports streaming audio. Sounds are used to :

- Support visuals
- Attract attention and interest
- Improve perceptions of quality
- Represent things not shown
- Provide feedback

### **Video**

When it comes to making an impact, video is right there at the top of the list. It takes a lot of computing power to incorporate video into a production, but it takes even more visual skills. Great ideas can be found by analyzing the videos shown on television. The video shown must be captured , edited , compressed and stored. A video file is an expensive resource. It can take up a huge amount of space on a hard drive. The images must tell their story quickly ,yet completely.



Video is used to :

- Show how things move
- Show that users cannot see directly
- Motivate users
- Show human behavior and emotions

### **Animation**

Animations plays a huge role in entertainment(providing action and realism) and education (providing visualization and demonstration ).Animation adds visual impact to the multimedia project. Many multimedia application for both Macintosh and Windows provide animation tools.

There are two main types of animation , which are used in multimedia applications. The first type of animation, 2D or linear animation, is the simple movement of objects on the screen, generally through linear translation. A multimedia development environment often provides the necessary tools to perform this simple animation procedure.

The original creation of animation, 3 D animation , however is very technically and artistically involved , as well as time consuming.

Three-dimensional objects are created using a mathematical model. Each object may be shown in various views , giving the user realistic sense of third dimension.

Besides 2D animation and 3D animation , there are also other special effects techniques in computer animation such as morphs and warping.

## Human – Computer Interaction

Running a multimedia presentation on a computer adds the capability of interactivity. Interactivity the viewer to navigate through a presentation in his or her own way and at their own pace. Users can jump from topic to topic and skip areas of little interest. Interactivity is the real key to an effective and successful multimedia title. It makes the program feels more personal to the user. Interactivity can be accomplished by linking the entire multimedia component- that is text, graphics, animation, audio and video- into a cohesive application.

## 2.6 The Components of Interactive Multimedia

Before developing the multimedia teaching package , allow me to examine the separate components of Interactive Multimedia and the interaction between media. Below are the separate components of Interactive Multimedia.

### Computer Assisted Instruction (CAI)

Computer Based Training employs technology to assist the instructor to instruct or to guide the learning program of individual users. Its main component consist of CAI and CMI , each of which may employ a variety of media. Computer Assisted Instruction (CAI) employs instruction modes of tutorials , review and practice and simulation.

### Computer Managed Instruction (CMI)

Computer Managed Instruction includes diagnostic assessment and prescriptive study assignments. Because of the leading role of developing and using CMI in Alberta, the topic will be accorded its own special separate section.



### Static Visual Display(SVD)

There are often called visuals or graphics .Their function is to provide a non-text based representation of some objects , process , concept or skill to be learned, although they are often accompanied by text. They do not move through space or time and vary in the amount of detail and realism they contain, from a simple line drawing created with a draw /point program to a photograph.

### Audio

Audio broadcasts information or data in a format which can be heard and may include instruction which is redundant with text or unique . It may also include warning sounds and sounds which are used to determine the stage of things , for example a badly- tuned automobile engine.

### Screen Design/Color

Screen design refers to how information is spatially organized for presentation to the learner. Some separate screen design issues are the use and placements of fonts , color and numerous design issues such as balance , borders and so forth. Color has been the object of numerous studies dating back to the middle of the century.

### Animation(Dynamic Visual Displays)

Animations are visual images which represent motion through space or time. Like SVDs , they may vary in the amount of detail and realism they contain , from a simple line drawings through to video.

They are usually created using special authoring software , such as Authorware, Director .Alternatively they may be video sequences which have been captured in analog or digital format. The set of motion visuals that includes animation and video has been called dynamic visual displays(Park & Hopkins ,1993)

### Multi-Channel Learning

Interactive Multimedia enables simultaneous delivery of instruction via several senses or channels. It might appear that the same information , delivered through multiple channels simultaneously (e.g. , visual and audio ) would enhance learning in a measurable way. Alternatively , could information from one channel interface with information from another channel and decrease learning. What if information delivered by one channel is in conflict with or irrelevant to information delivered via another channel ?The research and theory dealing with these and other topics is referred to as Multi-Channel research.

### Navigation

Navigation refers to the process of acquiring information from a rich multimedia data base which has obvious organizational pattern. The World Wide Web is an example of the latter. It is intuitive and attractive to believe that navigation as a learning system will result in significantly better learning than highly structured learning to date , there is little research in this area. The conclusion which seem to be emerging is that the effectiveness of navigation cannot be assumed present for all learning situations.

While research on navigation is quite new and as yet limited , it is expected to increase dramatically in the next few years.

### Instructional Television

Instructional television refers to the use of televised media, whether broadcast , analog tape or videodisc or digital imaginary which is employed is an instructional setting to supplement or supplant other forms of instruction. As with other forms of media, ITV originally required massive production facilities , personnel and budgets. With the advent of the transistor, microprocessors and chips, the equipment for capture and editing has become much less expensive and has migrated to the desktop.



The basics of good production have remained the same , although styles have been driven by the popular media, styles such as the short sound bite, rapid flashes of imagery, and special graphics effects.

### Interactivity

Except for the computer driven by microprocessor , multimedia are by and larger designed for transmission of information about content from a knowledge source to users (information transmission) and is incapable of interaction. There are many definitions of interaction but they generally require that two things be able to carry out activities which elicit a response from one another. Perhaps the highest level of interactivity is a series of activities which result in the student learning or processing information at a cognitive level which is higher than rote memorization.

## **2.7 Findings**

The fact-finding technique is the specific method for collecting data and relevant information pertaining to system requirements. There were 4 methods used to collect data and information on the requirements. They were Survey and Questionnaire, using the Internet, research on books, and the Document Room in Faculty Computer Science and Information Technology.

### **2.7.1 Survey and Questionnaire**

#### Survey

A survey was done to find the existing interactive multimedia teaching system existing in the market .The system available can be divided into CD- Rom based learning and web-based learning.

For CD-Rom learning, they weren't any systems that were available for teaching XML. So another system was analyzed just to note what is important in teaching a language and what important elements should contain in developing a good teaching package based on a CD-Rom.

For web –based learning, these sites were found on the Internet:

- a) TECFA's XML page – a multimedia teaching page that teaches XML.
- b) Working with XML- a course online which teaches XML
- c) XML Teaching – a flash interactive with animated tutorials

### Questionnaire

Users were asked four open ended questions:

- a) What was good about the CDRom (as a learning or as a resource)?
- b) What aspects did they like?
- c) What were the faults with the CDRom?
- d) How could it be improved?

In response to the all the question in general a majority of users cited the increased availability and accessibility of information. Half the users also mentioned the use of sound, video, graphics which they felt added interest, helped them to remember concepts and facilitated understanding of issues.

(showing numbers who made particular comments out of 50 respondents)

- availability/access. of information - 29
- audio/visual format helps- 25
- added interest -15
- improved learning tool compared to book-11



- aided understanding- 10
- variety of material- 9
- good for revision/elaboration- 7
- diversity of views presented- 7
- ease of navigation -7
- ability to work on own and at own phase- 6
- references to other material-1 6
- encouraged thinking- 5
- new and exciting- 5
- attractive- 3
- memorable/aided memory -3
- well researched/structured- 3
- comprehensive/integrated- 3
- liked interaction- 1
- helped research -other courses- 1

### 2.7.2 Using the Internet

Various search engines such as Yahoo , Google , Altavista ,etc was used to search for information. Specific keywords based on the type of information needed was used to search the necessary information. Below are the details of the sites visited :

### Site on interactive multimedia

- a) <http://www.gettowntown.edu/crossroads/mltmedia.html>

This site provides all the information about Interactive Multimedia. This site is very useful in defining interactive multimedia, why is it used, and the types of multimedia available. It also explains how multimedia can be incorporated into education context and technical requirements for using multimedia.

### Sites on web and interactive designing

- a) <http://tutorials.beginners.co.uk/view/cobrand/searchmiddleware/i/6>

This site provides useful tutorials authoring tools such as Macromedia flash , Macromedia Dreamweaver , Microsoft FrontPage , and so on . It also provide tutorials on JavaScript , HTML , ASP , Visual Basic , XML , the Internet , networking , CDROM and web development , web and CD-ROM marketing and so on .

- b) <http://www.med.monash.edu.au/informatics/techme/authorsoft.htm>

This site provides information on what is an authoring tool and considerations to be made when choosing software. It also lists a number of websites that discusses the issue of selecting the right software to develop a multimedia system. It gives detail explanation regarding certain authoring software such as Authorware , IconAuthor , Toolbook , and Macromedia Director.

- c) <http://www.macromedia.com>

This site was surfed to get information about design an development tools such as Dreamweaver 4 , Flash 5 , Fireworks 4 , Authorware , etc .



d) <http://www.adobe.com>

This site provides the information about web design tools such as Adobe WebCollection , Adobe GoLive , Adobe LiveMotion , Adobe Illustrator , Adobe Photoshop , etc . Adobe AfterEffects , Adobe Premiere , and Adobe Streaming Media Collection are examples of multimedia tools. Even though this are more web based designing , but it can also be incorporated into a CD-ROM based designing . The main difference between CD-ROM learning and web based learning is the linkage .

#### Sites on XML (Extensible Markup Language)

a) [Www/w3.org/XML/com](http://www.w3.org/XML/com)

This site is on the World Wide Web consortium on XML activity and information.

b) [www.xml.com](http://www.xml.com)

This site explains about the community resources and solutions, XML authoring tools , XML resources and interactive forums.

c) <http://java.sun.com/xml/tutorial>

This is an online manual that can quickly get us to speed writing XML code and XML- based software for end- user applications.

### **2.7.3 Research on Books**

There were various books used to gather information .Basically these books were used to gather more information on XML .Research was done to compare the contents of these books to prepare the teaching modules and also to learn more on this language. Other books were also used to learn more on authoring tools and their usage. Some of the titles of the books:

a) Just XML 2<sup>nd</sup> Edition

(Author: John E. Simpson ,Publisher : Prentice Hall)

This book explains about XML ,a step- by- step through all the basics of creating and maintaining your own customizable XML applications.

b) Learning XML  
(Guide to) Creating Self-Describing Data

In *Learning XML*, the author explains XML and its capabilities succinctly and professionally, with references to real-life projects and other cogent examples. *Learning XML* shows the purpose of XML markup itself, the CSS and XSL styling languages, and the XLink and XPointer specifications for creating rich link structures.

c) Title: XML Step by Step

XML STEP BY STEP clearly explains the basics of XML and shows both nonprogrammers and Web developers how to create effective XML documents and display them on the Web. The book takes a hands-on learning approach and focuses on the most practical techniques that you can use now, even if you don't have extensive technical knowledge. A CD-ROM that comes with the book includes XML examples for all hands-on exercises, plus extensive links to further information and resources.

#### 2.7.4.Document Room

Another source of gathering information is the Document Room which is located in the Faculty of Computer Science and Information Technology .Here some documentation on other students final year thesis that were done in the past years can be found. Some of this documentation were found relevant in providing information in developing this software. The documentation actually gives a brief idea on how to go about in developing a system from the very beginning , which is rather important in the developing this teaching package.



## 2.8 Development Tools

### 2.8.1 What is an Authoring Tool

For the purpose of this project research was done to know more about authoring tools. And from the information gathered, it is learnt that an authoring tool is simply something that helps to develop front ends for application, cut development time in doing so and increases programmers efficiency due to ease of user. However, a common definition would be a software package that has pre-programmed elements for the development of interactive multimedia software titles that can create multimedia presentation, training courses, on-screen productions, demo disks, guided tours, interactive kiosks, simulations and prototypes through scripting languages and icon-based construction environments.

Authoring tools provide true interactivity, not just button pressing and powerful control over media elements, including words, images, animation, sound and video. Authoring systems vary widely in orientation, capabilities and learning. There is no such thing as a completely point-to-click automated authoring system, some knowledge curiosity thinking and algorithm design is necessary. Authoring tools is actually just a speed up form of programming because you don't need to know the intricacies of a programming language or work an API but you do need to understand how the program works.

Advantages of Authoring Tools :

- Ability to do fast prototyping
- Ease of expanding the prototype to a full system
- Ease of use
- Built in multimedia capabilities
- Less need of programming expertise

2.8.2. Authoring Process

Authoring process involves choosing the right media elements and to synchronize them together to create a final multimedia production .To create a good multimedia application, users need to understand the multimedia process involved.

Steps involved in authoring a multimedia application includes :

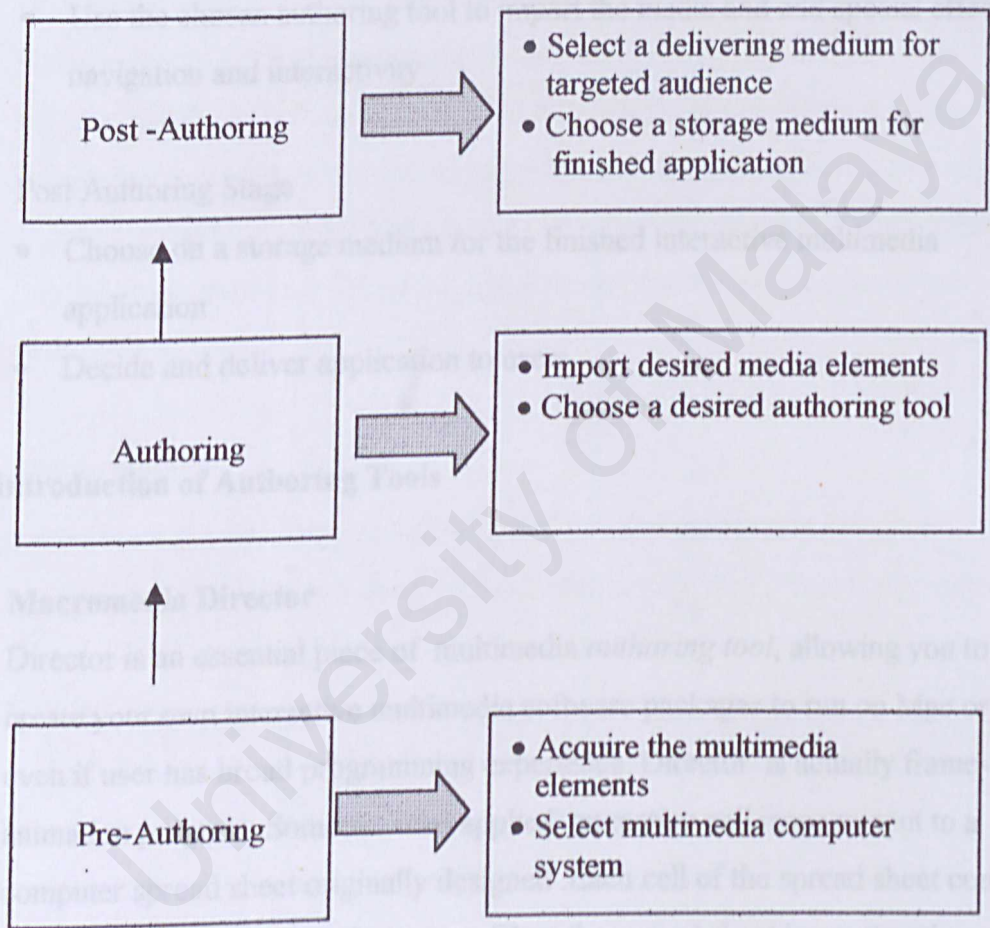


Figure 2.0 The Authoring Process

Source : Neo Main and Ken Neo TK , 1999



### Pre –Authoring Stage

- Select the multimedia computer and its component
- Acquire the multimedia elements and the necessary structure for creating and editing

### The Authoring Stage

- Choose an authoring tool to best suit your need
- Use the chosen authoring tool to import the media and add special effects , navigation and interactivity

### Post Authoring Stage

- Choose on a storage medium for the finished interactive multimedia application
- Decide and deliver application to users.

## 2.8.3. Introduction of Authoring Tools

### Macromedia Director

Director is an essential piece of multimedia *authoring tool*, allowing you to create your own interactive multimedia software packages to run on Mac or PC even if user has broad programming experience. Director is actually frame- based animation program. Someone who applied animation cell management to a computer spread sheet originally designed. Each cell of the spread sheet contained graphics into in the form of a bitmap. Then the spread sheet becomes a 'score'. When the score's cell was automatically sequenced using the computers built- in -clock , the bitmap images would sequence on the screen in a window. Animation also can be inserted which will be controlled by a spread sheet. Director is the main tool used for authoring CD-ROMs . It was originally designed as a package to integrate black and white animation and sound,

Director has evolved into a sophisticated tool allowing you to combine text, sound, animation, graphics and video, and to add user interaction using the high-level Lingo language.

Simple diagrams and texts, as well as sound can be supplementary product line – produced with Macromedia Director. However the used data or actor are imported by special programs.

This is then integrated together to form a complete work join. To really uncover the power of Macromedia Director, users must understand how to do command scripting. Macromedia's scripting language is called Lingo. With Lingo, users can write scripts to control objects, branching and even outside devices – like disc players. But the best thing about Director is that users can create run-time versions of users program, called projectors that users can distribute freely. Director started out on the Macintosh platform, but it has grown more robust throughout its evolution becoming a true cross platform multimedia authoring tool.

### **Authorware**

Authorware has flexibility and power, and handles multimedia elements appropriately for high-end development. Authorware also offers the potential for cross-platform development (that is developing a program that can run on both Windows and Macintosh machines) - but development must take place on the Macintosh and be transferred to a Windows machine. Most people believe that it is easier to develop in two platforms in parallel, than to use a 'cross-platform' tool. The latest version of Authorware offers the ability to transfer your creations to Shockwave (which enables you to play Macromedia Authorware (slowly) over the Web).

Authorware is therefore being promoted as an intranet tool. (An intranet is a network within an organization, as oppose to the Internet.) Authorware is also integrated with Director. However Director will not run on Windows 3.1, only on Windows 95, and 3.1 is still the University standard, although this will shortly change.



### **Authorware Academic**

Authorware Academic is a cheap (~\$150), cut down (only 300 icons in an application) version of Authorware Professional, designed for educational institutions. There aren't too many high-powered versions of Authorware Academic around, and it is similar to a prior product, Authorware Star, which did not handle multimedia or complex programming well, but was ideal for very simple products.

### **AimTech IconAuthor 7.0**

Icon Author is similar to Authorware in look and feel but different in terms of capability. IconAuthor's selling point is 'programming not required' and may be one of the most appropriate tools for those with little interest in the finer details of authoring. The product is released by Aimtech but is distributed in Australia by PPS technology who have excellent descriptions of the Icon Author product. IconAuthor is not good for programming. IconAuthor is also aiming at the Internet/HTML market, by allowing the transfer of IconAuthor to HTML.

### **Macromedia Flash**

Macromedia Flash (developed by Macromedia) is the leading vector graphics technology for designing high-impact, low-bandwidth web sites. Flash is the solution for producing and delivering high impact Web sites, as well as resizable and extremely well organized

SWF (ShockWave Flash) is the file format used by Macromedia Flash to deliver graphics, animation and sound over the Internet. These techniques are rapidly changing the way we play, work, or just browse online.

What are vector graphics? Vector graphics, which manipulate coordinates and mathematical formulas rather than pixel-by-pixel bitmaps, produce graphics files that are one-tenth the size of bitmaps.



Additionally, SWF can deliver animation, rich colors, sound, and interactivity. Moreover, this approach downloads faster, it's scalable (more on this below), and it boasts higher quality than other graphics formats.

### **Adobe PhotoShop 6.0**

Adobe® Photoshop® 6.0 software introduces the next generation of image editing with powerful new features that offer something for every user.

Delivering the broadest and most productive toolset available, Photoshop helps you explore your creativity, work at peak efficiency, and achieve the highest quality results across all media. As the industry standard for digital image manipulation software, Adobe Photoshop has revolutionized the photography and prepares industries and has provided commercial and fine artist with an exciting new medium for photographic editing. Adobe has integrated into Photoshop a design based upon traditional photo manipulation technique, where tools and processes directly correspond with those used in 'physical' photography.

Photoshop introduces features and enhancements that go far beyond the capabilities of the darkroom technician, thanks to digital technology; yet through an interface based on traditional technique, Adobe ensures a relevant, familiar, but powerful program environment. Covering a few of the general elements of the program will give users a better perspective on how it works, allowing us to uncover technicalities (tools and various command techniques) in more efficient terms.

### **Adobe Acrobat**

If you are interested in converting large quantities of written documents into electronic format Adobe's Acrobat technology has a lot of advantages. It remains the easiest and fastest way to convert word processed and graphic files into a cross-platform, network deliverable format. Acrobat's PDF format and Acrobat Reader technology are easy to distribute on CD-ROM, LAN or via the Web, and the PDF format retains a remarkable fidelity to the original document, including imported figures, scanned images, Greek characters, and formulas.



PDF files are larger, however, than HTML, the native language of the Web, but the creator has some latitude for compressing files further with a subsequent loss of image detail.

The creation of PDF files requires the purchase of Adobe's Acrobat. This package gives several options for creating PDF files for multiple users on both PC's and Mac's, but creating a PDF files can be as simple as printing a document.

The shortcoming of Acrobat is that users are required to download and install the Acrobat plug-in or helper application, Acrobat Reader. But many Web surfers already have Acrobat because it is so commonly used on the Web and it is frequently bundled with Web browsers. Searching through many Acrobat files on a Web site is possible if a web server's search engine supports that feature. Our main web server can index and search PDF files.

Acrobat also allows you to add hyperlinks within and between documents, embed audio or video files, and build truly interactive documents. The text within PDF files is now readily editable however, so updating hyperlinked PDF files can be laborious.

If you want to convert large quantities of highly formatted documents into electronic format Acrobat is an easy standard to recommend. Even though it requires special software on the users' end, its ease of use, fidelity and versatility make it a clear leader in the electronic document field.

### **Macromedia Fireworks 4.0**

Macromedia created Fireworks as a complete graphics applications aimed towards Web designers. It offers more advanced image control than existing packages, such as Photoshop or CorelDraw, for work that specifically relates to the Internet. Fireworks enables web designers to produce high quality images, export them in a variety of file formats while retaining excellent file size compression.



PDF files are larger, however, than HTML, the native language of the Web, but the creator has some latitude for compressing files further with a subsequent loss of image detail.

The creation of PDF files requires the purchase of Adobe's Acrobat. This package gives several options for creating PDF files for multiple users on both PC's and Mac's, but creating a PDF files can be as simple as printing a document.

The shortcoming of Acrobat is that users are required to download and install the Acrobat plug-in or helper application, Acrobat Reader. But many Web surfers already have Acrobat because it is so commonly used on the Web and it is frequently bundled with Web browsers. Searching through many Acrobat files on a Web site is possible if a web server's search engine supports that feature. Our main web server can index and search PDF files.

Acrobat also allows you to add hyperlinks within and between documents, embed audio or video files, and build truly interactive documents. The text within PDF files is now readily editable however, so updating hyperlinked PDF files can be laborious.

If you want to convert large quantities of highly formatted documents into electronic format Acrobat is an easy standard to recommend. Even though it requires special software on the users' end, its ease of use, fidelity and versatility make it a clear leader in the electronic document field.

### **Macromedia Fireworks 4.0**

Macromedia created Fireworks as a complete graphics applications aimed towards Web designers. It offers more advanced image control than existing packages, such as Photoshop or CorelDraw, for work that specifically relates to the Internet. Fireworks enables web designers to produce high quality images, export them in a variety of file formats while retaining excellent file size compression.



Fireworks surpasses other rivals as it also enables you to edit text, graphics and effects at any stage during production. Fireworks also simplifies the process of making JavaScript buttons with mouseover effects, all combined with automatic HTML generation.

This products is obviously aimed squarely at web designers who are currently restricted by the limitations of Photoshop and the endless workarounds that are necessary to convert images accurately to the web.

Fireworks, in addition to greater palette features, has well-developed type controls and vector tools for creating and editing lines, shapes, and Bezier paths.

Fireworks goes one better than most by offering paint-style brush strokes, texture fills and special effects such as bevels and drop shadows. All objects can be scaled, skewed, rotated and made transparent. Fireworks, again surpasses Photoshop by enabling you to modify effects at any stage and update them onscreen. When you change an object, Fireworks retains and updates the effects.

The Painting Tools are similar to current paint programs, with convenient points and dragging control handles. Fireworks improves on the rivals, as after you've defined a vector shape, you can select a bit-map paint style to stroke the path. In addition the original vectors still remains editable. The paint styles also includes oil brushes, charcoals and watercolors, as well as the standard solid strokes.

All normal drawing functions are provided, including grouping, layering and a snap-able grid. Missing are Boolean controls and the ability to lock or hide individual shapes. Drawing tools include standard primitives as well as freehand, text and Bezier pen - with a welcome modification to the Pen tool: When you click an initial point and drag out a handle to start a curve, the program displays a rubber-banding curve that shows what your shape will look like when you click the next point.

In addition to fill and stroke, any object can have a Live Effect added to it. These effects include inner and outer bevels, drop shadows, halos and glows, each with a number of parameters that can be changed at any time. You can apply more than one effect to an object by grouping the object with it's effect, add another effect, group, ... etc.



Fireworks does a very good job in converting shapes into a bit-map. It can also import bit-map images created in another programs, such as Freehand, and edits them using its editing tools. There are also a sleuth of brush, rubber-stamp, graphic-primitive and bucket tools, as well as blur and sharpen filters. Any Adobe Photoshop compatible filter can also be applied.

Fireworks shows its superiority over rival packages with its excellent image map tools. These tools allow you to create very powerful server-side image maps. The URL toolbar allows you to draw primitive shapes and then assign URLs to each. Navigation images can be created in a few simple steps. Fireworks also includes tools for creating simple animated GIFs.

### **ToolBook II Instructor**

ToolBook II Instructor is the 'complete' ToolBook. Although some templates are provided, it is necessary to learn some of the ToolBook scripting language.

ToolBook programming is based on the paradigm of a book. Each file is considered a book, consisting of separate screens called 'pages'. The 'pages' are filled with 'objects' such as 'fields' which can contain text, graphical objects like circles or lines, and media objects like bitmaps or digital video.

ToolBook is powerful in terms of its native (somewhat 'object-oriented') scripting power - and is therefore appealing to programmers. There is a sizeable learning curve and it takes even experienced people a while to become proficient. ToolBook also promotes itself as offering interactive Internet capability, however, like all other authoring-package-to-HTML converters (such as PowerPoint and Authorware), it is only applicable to intranet situations. (An intranet is a network within an organization, oppose to the Internet.)



### **Toolbook II Assistant**

ToolBook II Assistant is designed for those who are not interested in learning how to program. Assistant provides a number of templates of computer assisted learning packages such as multiple-choice quizzes, tutorials and so on. Extra features are provided by assembling 'widgets' - small building blocks of software - on the page to create your desired program. Scoring, responses to answers and so on, are all part of the basics provided. Obviously it is a restricted environment.

### **Microsoft FrontPage**

The Microsoft FrontPage is a tool that can be used as a web site creation and management tool gives everything that is need to create and manage exactly the site whatever creating a personal Web page or corporate Internet or intranet site. It is easy to use and produces good result. It allows you to insert graphics, create tables, forms and either advanced HTML features, in an intuitive fashion. It gives you a graphical overview of your web site, showing links between all your documents and installs your own personal Web server.

## **2.8.4 Scripting For Multimedia**

A complete understanding of the component parts of the content and the places where there may be occasion for the user to move sideways or diagonally through the content as well as up and down. It is also important to remember that an interactive viewer has control over the order in which the information is presented – there is no definite beginning, middle or end.

Alternatively, the program might need to track what the user has done and tailor access to the rest of the information accordingly. The content is intertwined with media types and techniques because they affect each other.

The main sections of content in an application might involve the use of different media, and may well have had different people script them.

This might mean that they have component integrity but not cross-component integrity. Another problem might be that each section has different weighting—one could be 2 minutes long, another 20 minutes long. Unless there is a good reason, different treatment of the depth of the content risks leaving the user puzzled and unsatisfied. The purpose of the communication affects the breadth and depth of the content. People absorb different types of information in different ways and for different reasons.

Multimedia scripting is complicated and demands a complex set of skills.

These include :

- Making decision on what material is relevant,
- Deciding which media would be right for the purpose,
- Scripting various levels of materials in different media,
- Providing the cohesion between the level by providing navigation diagrams /flow charts,
- understanding the user and the implications of user choice on the structure of the material,
- being able to communicate with and instruct the programmer,
- keeping the vision of the whole dissecting it into fragments,
- organizing the material into a detailed design document that the team can understand.



## Lingo Scripting

Lingo, Director's scripting language, adds interactivity to a movie. Often Lingo accomplishes the same tasks-such as moving sprites on the Stage or playing sounds-that you can accomplish using Director's interface.

Much of Lingo's usefulness, however, is in the flexibility it brings to a movie. Instead of playing a series of frames exactly as the Score dictates, Lingo can control the movie in response to specific conditions and events.

For example, whether a sprite moves can depend on whether the user clicks a specific button; when a sound plays can depend on how much of the sound has already streamed from the Internet.

*Behaviors* are pre-existing sets of Lingo instructions. Attaching behaviors to sprites and frames lets you add Lingo's interactivity without writing Lingo scripts yourself.

If you prefer writing scripts to using Director's interface and behaviors, Lingo provides an alternative way to implement common Director features; for example, you can use Lingo to create animation, stream movies from the web, perform navigation, format text, and respond to user actions with the keyboard and mouse. Writing Lingo also lets you do some things that the Score alone can't do. For example, Lingo's lists let you create and manage data arrays, and Lingo operators let you perform mathematical operations and combine strings of text. We can also write our own behaviors that perform tasks beyond those possible with the behaviors that already have available.

## 2.8.5 Other Tools

### CoolEdit 2000

- Record files from a CD, keyboard, or any signal played through your sound card.
- Edit and master audio with professional digital tools.
- Read and write MP3 using Fraunhofer-based codecs.

- Convert files from one format to another--Cool Edit 2000 is the ultimate audio translator
- Touch up files with functions like Filter, Amplify, Compress, Stretch, and Noise.
- Add fantastic effects with Reverb, Delay, Echo, Flanger, Distortion, and Envelope Generate Silence, Noise (White, Pink, or Brown), DTMF Signals, and Tones.
- Create sound effects of your own.
- Analyze audio with Spectral View and Frequency Analysis.
- Convert sample types with noise shaping.
- Use the Cue List and Play List to cue up and play the segments you specify.
- Create named Presets to store your favorite processing function settings.
- Create Scripts for automated processing and batch operations.
- File formats include: MP3, WAV, AIF, AU, VOX, and export to RealMedia G2

### **SoundForge 5.0**

Sound Forge is Sonic Foundry's award-winning two-track digital audio editor. Sound Forge includes a powerful set of audio processes, tools, and effects for manipulating audio. This one-of-a-kind application is perfect for audio editing, audio recording, effects processing and media encoding. Combine Sound Forge with any Windows-compatible sound card to create, record, and edit audio files. The clean and familiar Windows interface makes editing a breeze. It also has built-in support for video and CD burning and can save to a number of audio and video file formats, including WAV, WMA, RM, AVI, and MP3.



2.8.6 Comparison On Authoring Tools

Two of the famous Authoring tools Authorware and Macromedia Director were compared . And the results were:

Authorware	Director
<ul style="list-style-type: none"><li>• Information navigation</li><li>• Business, Educational applications</li><li>• Kiosks (information provision)</li><li>• Prototyping business applications</li><li>• If it were a magazine it would be: FACE<ul style="list-style-type: none"><li>- trendy yet easy to read and functional</li><li>- the magazine has a holistic design philosophy</li></ul></li><li>• Education/Business use growing</li><li>• Easy to use</li><li>• Flow chart -type- metaphor</li><li>• Focus on: information navigation &amp; product</li><li>• Bad: can be critic when used in anger</li><li>• Good: information navigation,  ease of use, indexing,  education for CBT and  PSS/MPSS in business</li></ul>	<ul style="list-style-type: none"><li>• In your face multimedia</li><li>• Magazines, Games</li><li>• Kiosks (full multimedia)</li><li>• Experimental advanced interfaces</li><li>• If it were a magazine it would be: WIRED<ul style="list-style-type: none"><li>- full on graphics - difficult to navigate</li><li>- every page designed for FULL impact</li></ul></li><li>• Extensively use in the Multimedia industry (# 1 app)</li><li>• Complicated to use</li><li>• Film (actors, story line) metaphor</li><li>• Focus on: product</li><li>• Bad: complicated, heavy on scripting</li><li>• Good: multimedia handling,  2D animation</li></ul>

Two more other Authoring Tool which is more popular in image –editing. These software's are used to edit image that are used in a production. The two softwares that were compared are Adobe Photoshop 6.0 and Macromedia Fireworks 4.0. From the research there are certain features that Photoshop has that Fireworks doesn't have.

They are:

1. Full-featured image editing

Fireworks is adequate for buttons , banners , and such , but it falls short when you need to create more complex , photographic imagery. Having much more precise tools for image masking and color correction, Photoshop is indispensable for realistic retouching and composting of photos.

2. Faster , more flexible vector editing

Renowned for its industry-leading image editing, Photoshop now provides a seamless way to combine raster images with crisp ,vector edges . Unlike Fireworks , Photoshop 6.0 software offers live Pathfinder operations to add , subtract , or otherwise combine simple shapes without losing the individual paths . Photoshop also provides a wide variety of customizable shape libraries in logos , buttons , and other designs .

3. Wider range of creative effects .

The controls for such effects as drop shadows and bevels are much more extensive in Photoshop then in Fireworks , allowing design professionals more creative freedom so as to avoid cookie-cutter design . For example , Photoshop has twice as many drop shadow controls as Fireworks , and more than four times as many bevel controls. Plus, Photoshop software's collection of more than 95 filters goes well beyond the handful of effects available in Fireworks .



#### 4. One-click rollover creation

Creating rollover effects in Photoshop is much easier process required in Fireworks . The new rollover style feature in Photoshop allows you to create JavaScript rollovers as simply as clicking on a rollover style in the style in the styles palette . The selected layer is automatically sliced , and the associated effects are applied to each rollover state . Save your own custom rollover states just as easily , making it easy to apply the same effects to multiple buttons .

### 2.9 Comparison of Existing Systems

#### 2.9.1 TECFA 's XML page

A multimedia teaching page that teaches XML .This page is divided into different stages and parts.It is divided into six main parts. They are

##### **1.XML @ TECFA**

§ 1.1 Teaching XML @ TECFA § 1.2 Software installed at Tecfa § 1.3 Plans for R&D

##### **2.Documentation & Validation**

§ 2.1 On-line Validation § 2.2 Manuals § 2.3 Specifications, standards and official propaganda § 2.4 FAQ's § 2.5 On-line Mags § 2.6 NewsGroups and Mailing Lists § 2.7 General Tutorials § 2.8 Other Tutorials (XML Schema, Xlink, etc.) § 2.9 Introductory Articles § 2.10 Other Articles

##### **3.XML for programmers**

§ 3.1 Tutorials for programmers § 3.2 Java APIs § 3.3 Libraries for programmers (parsers, etc.)

##### **4.Software**

§ 4.1 Indexes for tools / software § 4.2 Browsers and plugins (client-side XML) § 4.3 On-line Tools § 4.4 Editors, editor libraries, and other tools § 4.5 XML Application / Publication Frameworks § 4.6 Server-side XML=>HTML § 4.7 Converters and other tools § 4.8 HTML in XML / HTML=>XML transition § 4.9 XUL

**5.DTD's, schemas, stylesheets, example XML etc.**

§ 5.1 Indexes § 5.2 Text § 5.3 XML Contents and DTDs § 5.4 Other DTDs § 5.5 General articles about schemas § 5.6 XML in education

**6.XML Links**

§ 6.1 Some major general indexes § 6.2 Indexes in french / en français

This page has a broad view on XML rather than concentrating on one part in XML.

**2.9.2 Working With XML**

The Java™ XML Tutorial, is an online manual that can quickly get you up to speed writing XML code and XML-based software for end-user applications. Using step-by-step walkthroughs and lots of code examples. The Tutorial is divided into sections that cover the following major topics

- explains the basics of XML and gives you a guide to the acronyms associated with it. It also provides an overview of the Java XML APIs you can use to manipulate XML-based data.
- **Serial Access with the Simple API for XML (SAX)** walks you through the process of creating an XML file and tells you how to read it sequentially, walking you through the callbacks the parser makes to event-handling methods you supply.
- **XML and the Document Object Model (DOM)** explains how to create and manipulate a hierarchy of data objects so you can randomly access it and modify its contents.
- **Using XSLT** shows how to transform an XML structure into a different format. This is also the API you use to write an XML file from a tree of objects in memory, and to create XML from an arbitrary document structure.



### 2.9.3 XML Teaching

A flash interactive with animated tutorials .This online package contains flash interactive interfaces that teaches users about XML.It contains questions and answers that can help users understand XML better.

### 2.10 Project Outcome

After comparing the other teaching tools that are available, Learning XML will have better features such as:

- Comprehensive
- Interesting interfaces
- Contains movie clips, comprising animations,graphics, digitized video and audio
- Help file that helps users to get around better
- User friendly

## CHAPTER SUMMARY

This chapter covers the project background study that is an introduction to XML and the differences between HTML and SGML. It also explains about the terms, the elements and the importance of multimedia and its usage in the package. This chapter also has findings of the project on how information and requirements were gathered. The introduction on authoring tools and the comparison for these tools were done in this chapter. The comparison on the existing system is also explained in this chapter

## LIST OF REFERENCES

- 1) Simpson, John E ,Just XML 2<sup>nd</sup> Edition,Prentice Hall
- 2) Learning XML , (Guide to) Creating Self-Describing Data ,O Reilly Publications
- 3) XML Step by Step, Prentice Hall ,2000
- 4) <http://www.gettown.edu/crossroads/mltmedia.html>
- 5) <http://tutorials.beginners.co.uk/view/cobrand/searchmiddleware/i/6>
- 6) <http://www.med.monash.edu.au/informatics/techme/authorsoft.htm>
- 7) <http://www.macromedia.com>
- 8) <http://www.adobe.com>
- 9) [Www/w3.org/XML/com](http://www.w3.org/XML/com)
- 10) [www.xml.com](http://www.xml.com)
- 11) <http://java.sun.com/xml/tutorial>



CHAPTER 3. METHODOLOGY

3.1. An Introduction to Systems Analysis

The primary goal of the development of the whole package. This analysis will help in the choice of the type of tools to use as well as the functional and non-functional requirements. Some of these are part of the requirements analysis and modeling of the system.

The process of analysis involves the following procedures:

- a) System Analysis
- b) Functional Analysis
- c) Modeling
- d) Specification

The objective

• Analysis

• Design

• Implementation

• Maintenance

• Evaluation

• Documentation

• Communication

• Management

• Organization

• Personnel

• Resources

• Environment

• Constraints

• Risks

• Opportunities

• Challenges

• Solutions

• Results

• Conclusions

• Recommendations

• Appendixes

# CHAPTER 3 METHODOLOGY

Before a new system can be designed, it is necessary to identify the problem that needs to be solved in order to develop a solution.

For this methodology to be effective, the following problem must be solved before starting the design process:

- (i) The problem should be clearly defined.
- (ii) The problem should be understood by the user.
- (iii) The problem should be understood by the management.

## 3.2. Problem Analysis and Synthesis

In this analysis of the problem, the user is required to break down the problem into smaller parts so that the problem will be easier to be understood and can be solved.

## CHAPTER 3 – METHODOLOGY

### 3.1 An Introduction on System Analysis

The system is part of the development of the whole package. This analysis will help in the choice of the type of tools to use as well as the functional and non – functional requirements. Both of these are part of the requirement analysis and requirements specifications.

The process of analysis involves the following procedures :

- a) Problem Identification
- b) Evaluation and Synthesis
- c) Modeling
- d) Specification

The objective of this analysis is to:

- Analyze the various tools available and choose the most suitable
- Determine the different models
- Determine the hardware and software requirements

#### a) Problem Identification

Before a new system can be built , we must identify problems that need to be solved in order to ensure its success.

For this multimedia teaching package , the following problem must be solved before starting this project:

- (i) What type of information should be put into the project
- (ii) How to present the information to the user.
- (iii) How to put information into the multimedia teaching package

#### b) Evaluation and Synthesis

In this state analysis of the problem need to be done by breaking the problems into smaller part so that the problem will be easier to be understood and can be solved .



The following problems are the system requirements that must be divided :

- (i) where to collect the information
- (ii) mode of presenting the information ( text, graphic, animation ,etc)
- (iii) what software will be used
- (iv) what are the requirements when the package is running

### **c)Modeling**

A model was created to gain a better understanding of the actual entity to be built. The model focuses on what the system will run on usually a graphical notation ( such as data flow diagram ) allow one to have a clear idea when designing the system.

This system will have six different module. This has benefit s because if mistake is made in one of the module , it does not effect the other modules and problem can be solved quickly because the separate module is small.

Another benefit is after one of the module is finished , it is easy to evaluate and finally integrate with other modules.

### **d)Specification**

The requirements specification is a complete picture that defines what the system should do. It would be used in the system design and system analysis .

## **3.2 Overview of the methodology process**

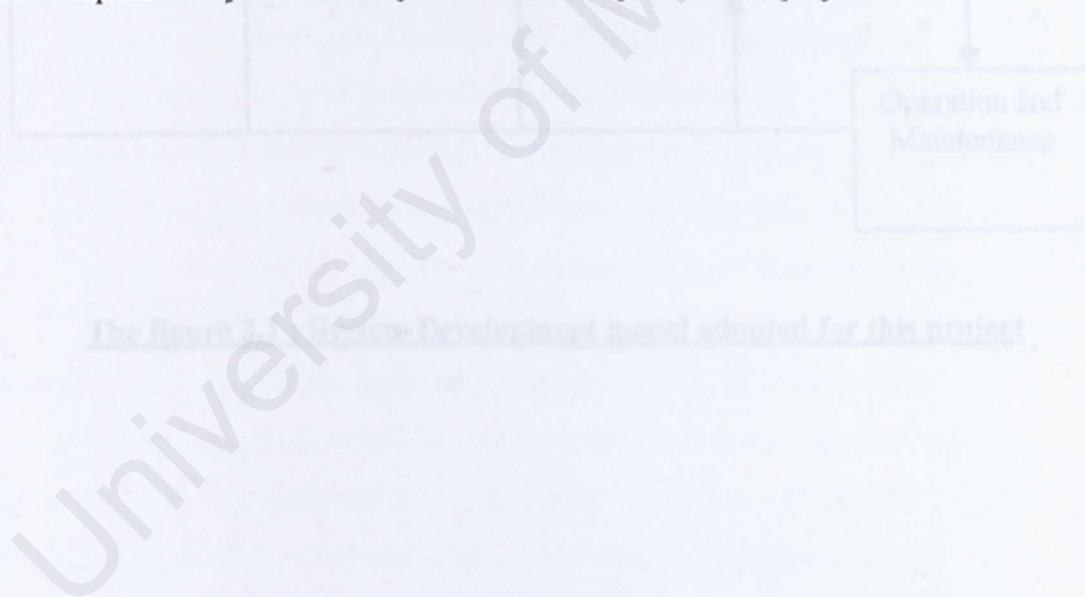
Methodology is a process that is used to develop a system. It is important to understand the nature of a program when developing a system. The services that the system should provide and the constrains under which it must operate must be established from the beginning of the system development process.

There are many advantages that can be achieved by doing research early in the system development process.

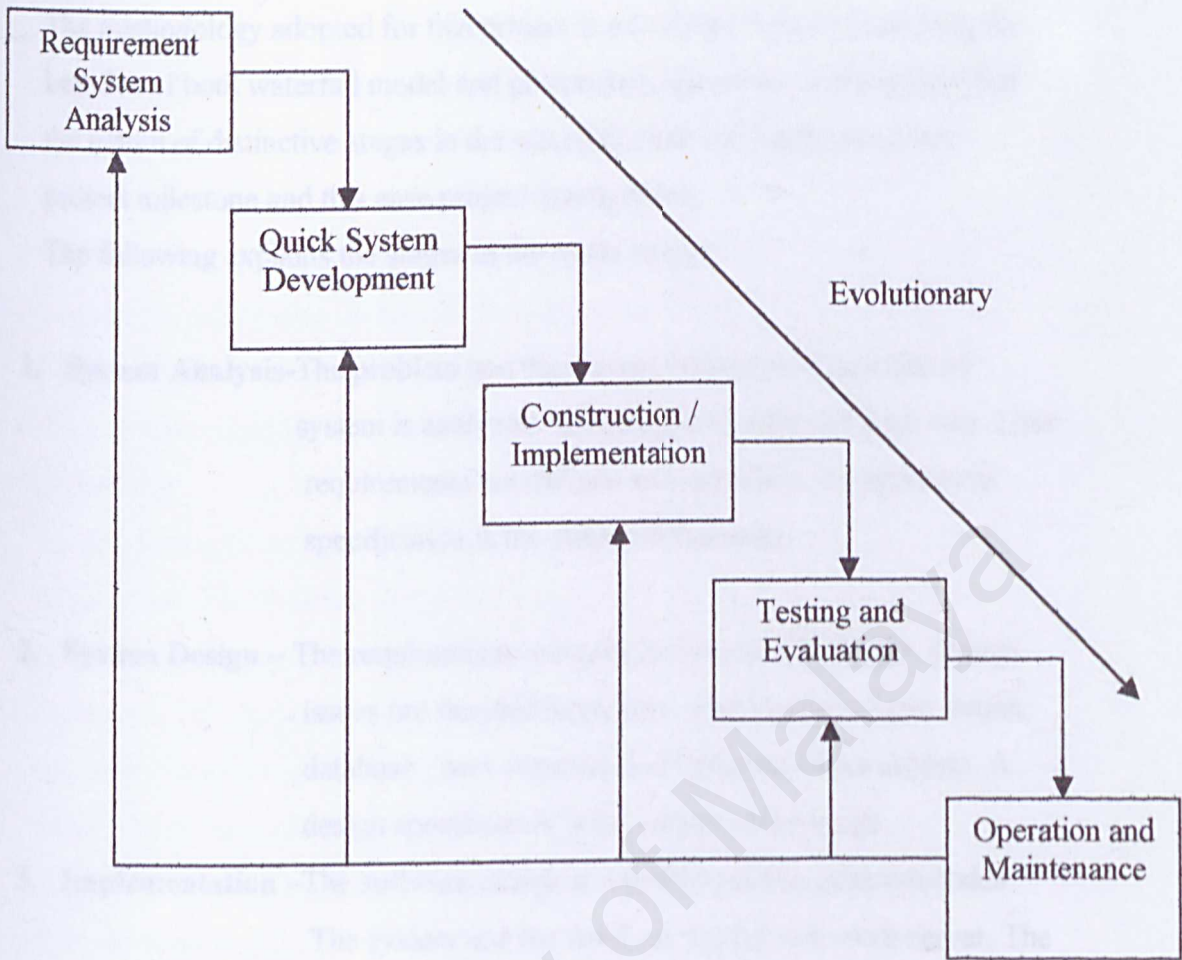
The developer is able to get through understanding of the requirements of the proposed system and will ensure that objectives of the system are fulfilled. Early research also gives the developer a clearer and better view of the system features to be incorporated through analysis of currently existing system.

### 3.3 System Development Methodology

Each methodology has its own strength and weakness. The traditional Waterfall Model is most useful because it provides a framework of stages to ease project management and it has been well used and tested in the industry. To solve the inherent problems, prototyping needs to be integrated to recognize the iterative nature of the software development model adopted for this project. The figure 3.1 below depicts the system development model adopted for this project.







**The figure 3.1 : System Development model adopted for this project**

The methodology adopted for this project is a modified version combining the benefits of both waterfall model and prototyping approach . It recognizes that the nature of distinctive stages in the waterfall model will help determine project milestone and this ease project management.

The following explains the stages in the methodology :

1. **System Analysis**-The problem and the current manual or computerized system is analyzed. Information is collected from user. User requirements are defined and specified. A requirement specification is the output of this stage.
2. **System Design** – The requirements are mapped into system design. Design issues are decided upon here. This involves input, output, database , user interface and designing other aspects. A design specification is the output of this stage.
3. **Implementation** –The software design is transformed into program codes. The system and the database are put onto web server. The output of this stage is an operational system readily to be tested.
4. **Testing** – Validation and verification are carried out . Verification seeks to match the system with user requirements. Testing involves unit testing, integration testing, system testing and acceptance testing. The outcome from this stage is a validated and verified operational system.
5. **Operational and Maintenance** –The system is put into practical use. Changes are made to the system whenever there is need to correct error , to perfect the system and to adapt the changes in the requirements of environments.



An evolutionary prototype is adapted to compensate the limitations of the waterfall model. In the stage of system design, it seeks to evaluate different design approaches. During the stage of implementation, it is to evolve into a full scale final system.

Since the prototype is evolutionary in nature, it is built knowing that it will become the full system ultimately. Hence the prototype is built in a modular fashion. Codes are reused whenever possible. Software quality is stressed and documentation is done throughout programming of the prototype.

### 3.3.1 Justification

By developing prototype functions instead of huge specifications, the user is given a real-life feedback of the end result. Accomplished through forth generation languages, this leads to a quick system design intended for full system testing, and can be enhanced over time until it is user-acceptable and optimal in terms of operations. Thus, such languages offer a unique approach that replaces the traditional system development steps in reaching a satisfactory solution in terms of end user computing. The method can be used all the way from the actor or function level to the node structure level to help decide if a system is feasible.

### 3.3.2 Strength

- By using the prototype process, customers or designers are able to try out a requirement before agreeing to it.
- User can discover requirement errors or oversight early in the software process. Requirement validation process consist of seven factors : correctness, consistency, completeness, realistic, needed, verifiable and traceable.
- Misunderstanding between software developers and users may be identified while the system functions are demonstrated.
- User services which are difficult to use or unclear may be found and corrected.
- Incomplete and/or inconsistent requirements may be found and corrected.

- In a short time , a working system is available to demonstrate the capabilities and usefulness of the application to management; however , this working system is limited.
- The specification for a production – quality system can be derived from the prototype.
- Prototyping can be viewed as a risk reduction technique. Experiments have shown that prototyping reduces the number of problems connected to requirement specifications and the overall development cost may be lower if a prototype is developed.

### 3.3.3 Weakness

- Progress is difficult to measure since it is not visible. Also if the systems are developed rapidly , documentation is usually not created to reflect each version of the system.
- The system may be poorly structured ,. The constant change is harmful to the software structure.
- Special skills are often required. Small teams of talented and motivated individuals are needed to succeed with prototyping.
- Major technical problems revolve around the need for rapid software developments.
- Prototype users or evaluators may be pressured by managers to draw quick conclusion concerning the prototype.

## 3.4 Software Development Tools Analysis

### 3.4.1 Authoring Tools

#### Macromedia Director 8.0

Macromedia Director 8.0 is one of the most common authoring tools. Originally for Macs, it is now available for all Windows-based platforms. Director uses the metaphor of a theatre, with a 'stage' and a 'cast' of objects. Director is less powerful than Authorware or Toolbook as it is less of a programming environment, but it is worth considering for highly graphical



computer-assisted learning packages with straightforward interactions. n  
Animation capabilities are very good.

Other commonly used multimedia authoring packages are Asymetrix Toolbook, Macromedia Authorware, Apple Media Toolkit, Supercard. These packages have some of the features below; Director has all of them.

- Excellent animation possibilities - allowing you to specify which layer a graphic is in relative to others, and removing the drudgery of setting up animations in other packages.
- Cross-platform compatibility - allowing playback on Mac or PC, given certain limitations. It is now being written for Silicon Graphics workstations too.
- Object orientation allows greater levels of control than an icon-based procedural package such as Authorware.
- Ability to set up a multi-window program, fully utilizing the potential of Windows and System 7, and allowing messages to be sent between windows.
- Director can be extended using X Objects, code segments written in C++ or another lower level language to address Windows (or the Mac OS) directly.
- Royalty free distribution.
- Huge user base ensuring future development.
- Exciting future as *Shockwave*, the Director engine shortly to be integrated with the Netscape Internet browser.
- Cost effective. Much cheaper than Authorware if you want to sell your packages.

### Macromedia Flash 5.0

Macromedia Flash allows to create a Web experience that is more attractive, more compelling, and compatible with more browsers than with any other Web solution. Dynamic HTML (DHTML), Java , and other advanced web design web formats are not reliable alternatives, since they're either compatible or inconsistent with different browsers. With Macromedia Flash Players pervasive penetration, wide availability, scalable vectors, animation, sound and more Macromedia Flash provide the high production quality that attracts users and

make them impressed. Macromedia Flash is light weight and powerful, Macromedia's Flash helps solve the bandwidth dilemma faced by developers. Sites use this technology to create buttons and controls that give animated feedback which will make the production more interesting.

## Adobe PhotoShop 6.0

Adobe® Photoshop® 6.0 software introduces the next generation of image editing with powerful new features that offer something for every user. Delivering the broadest and most productive toolset available, Photoshop helps you explore your creativity, work at peak efficiency, and achieve the highest quality results across all media Adobe Photoshop 6.0 software introduces the next generation of image editing capabilities with powerful new features that offer something in every user.

It can:

- **Expand Beyond Pixels**  
Photoshop 6.0 redefines traditional image editing with integrated vector drawing tools and new layer design features that greatly enhance your creative options.
- **Produce superb Graphics**  
Photoshop 6.0 expands its web toolkit to provide more timesaving, Production-oriented features including Adobe Image Ready 3.0 for advanced multimedia production
- **Master the power of Photoshop quickly**  
Photoshop 6.0 is designed to make it faster and easier for everyone to get to work and its comprehensive toolset more fully.



### 3.4.2 Programming Tools

Lingo, Director's scripting language, adds interactivity to a movie. Often Lingo accomplishes the same tasks-such as moving sprites on the Stage or playing sounds-that you can accomplish using Director's interface.

Much of Lingo's usefulness, however, is in the flexibility it brings to a movie. Instead of playing a series of frames exactly as the Score dictates, Lingo can control the movie in response to specific conditions and events.

For example, whether a sprite moves can depend on whether the user clicks a specific button; when a sound plays can depend on how much of the sound has already streamed from the Internet.

*Behaviors* are pre-existing sets of Lingo instructions. Attaching behaviors to sprites and frames lets you add Lingo's interactivity without writing Lingo scripts yourself.

If you prefer writing scripts to using Director's interface and behaviors, Lingo provides an alternative way to implement common Director features; for example, you can use Lingo to create animation, stream movies from the web, perform navigation, format text, and respond to user actions with the keyboard and mouse. Writing Lingo also lets you do some things that the Score alone can't do. For example, Lingo's lists let you create and manage data arrays, and Lingo operators let you perform mathematical operations and combine strings of text. We can also write our own behaviors that perform tasks beyond those possible with the behaviors that already have available.

### 3.4.3 Other Tools

#### SoundForge 5.0

Why use SoundForge 5.0?

- Powerful editing features

Sound Forge provides a combination of over 35 audio effects and processes with more than 200 presets. It includes dozens of editing features including: Cut, Paste, Move, Delete, Mute, Reverse, Crossfade, Trim, Normalize, Fade, Pan, Resample, Enhance, Insert Silence, and more.

- **Amazing effects**  
Sound Forge includes 20 DirectX Audio Plug-Ins, including, and, which can be used for acoustic simulation and microphone modeling.
- **The entire arsenal of effects includes Amplitude Modulation, Chorus, Delay, Distortion, Flange, Gapper, Noise Gate, Pitch Bend, Reverb, Vibrato, Time Compression, Wave Hammer™ and more.** Sound Forge also includes three Eqs—Graphic, Parametric, and Paragraphic and supports DirectX Plug-Ins from third party applications.
- **Support for many file formats**  
Sound Forge supports a wide range of audio formats, including 12 import formats and 17 export formats, such as WAV, Windows Media™ Audio, RealAudio® and MP3.
- **Cost value**  
Sound Forge includes more than \$800 of free software, such as XFX1™, including Learn more about these products by clicking on the hyperlinks that are provided.
- **Ease of use**  
Sound Forge provides simple drag-and-drop operations and a customizable interface, allowing you to build projects the way you want. Sound Forge also supports standard Windows® keyboard commands and mouse shortcuts.
- **Protected work environment**  
Sound Forge includes auto crash recovery, which allows you to recover lost work after a power failure or system crash. Sound Forge also includes a powerful undo/redo history feature, which allows you to see your entire work history at a glance and undo unwanted edits.

### Microsoft Access 2000

Access is a database platform which makes it possible to develop database integrated ASP websites. Because all the information in Access databases is contained in a single file, it is easy to upload or download the entire database over the FTP or the web.



Access allows the user to link together data stored in more than one file. It stores information in an easily retrievable form. It can store information such as text, numbers, dates, currency, pictures and sounds. S well as being able to store data , it allows information to be selected easily and quickly.

The summaries of the information selected can also be printed.

There are important things that should be considered when setting a up a database :

- type of information that needs to be stored
- types of information that needs to be retrieved
- who the data is intended for and how others will use it
- whether certain parts of the data is restricted to certain users only
- who is allowed to change or add data
- If the data refers to actual people, it may need to be registered under the Data Protection Act

Why use Microsoft Access ?

- Improved wizards.  
The wizards help you create tables, forms, and queries and are even more intuitive to use.
- Name AutoCorrect.  
When you rename a field (column) in a table, the change is automatically updated in all dependent tables, queries, and forms.
- Drag and Drop to Excel.  
You can export data from Access 2000 to Excel 2000 simply by dragging the Access 2000 objects into Excel 2000 (tables, queries, and so on). This feature is convenient when you maintain information on your students in both Access 2000 and Excel 2000 and you need to transfer some data from one file to another.
- **Compact on close.**  
Databases can grow in size quickly and take up space on your hard disk. Access 2000 automatically compresses a database when the file is closed to reduce the file size. This is especially useful when space

on the hard disk is limited and you are storing databases for a number of classes.

- Report snapshot.

You can create snapshots of Access 2000 reports that can be distributed to a disk, printer, Web page, or e-mail.

- Data access pages.

Creating Web pages from the objects in your database is a simple process using this new feature in Access 2000. Now your students can view and add information to a class database easily from your school intranet or from home.

### 3.5 Requirement Analysis

Requirement analysis is an important process the acceptability of the system. After it has been delivered it depends on how well it meets the specified needs and support the work to be automated. If the analyst does not discover the real requirement, the delivered system is unlikely to meet their expectation. A requirement is a feature of the system or a description of something the system or a description of something the system is capable of doing in order to fulfill the system's purpose. As for this project, the requirements were gathered through research on the Internet and books, analyzing on the results from the survey conducted, and review of the existing system.

#### 3.5.1 Requirement Elicitation

Requirement elicitation is a critical part to determine the requirements of the projects. A variety of techniques must be used to determine the users need and what they really want in the system. The techniques that were used are:

- i) Internet Research-The research was done to find some of the multimedia teaching package that was available online.
- ii) Interviewing- Some software engineers were interviewed since XML is getting more popular and can be integrated with the languageJava. This was to understand on what are the requirements that will be needed and have to added.



From what have been gathered ,requirements can be separated into three categories :

- Requirements that absolutely must be met
- Requirements that are highly desirable but not necessary
- Requirements that are possible but could not be eliminated

### 3.5.2 Functional Requirements

A functional requirement describes an interaction between the system and its environment .It is statements of services the system should provide , how the system should react to particular inputs and how the system should behave in particular situations. It also describes how the system should .Some of the requirements are :

- **Notes Module**

For each topic there will be facts and keywords that are useful for future reference. It will also contain a brief overview in what is important in that chapter.

- **Interactive Example Module**

In this module there will be example on certain important processes that occurred. It will be an interactive module so users will be keen to look at the examples.

- **Tutorial Module**

A specific module of tutorials for each chapter that is thought will be prepared. This tutorial will be based on the lessons that were thought. Users can choose on which topic they will want to do the tutorial. Answers for each question will be provided. This is to help them understand each chapter better.

### 3.5.3 Non Functional Requirement

Non-functional Requirements is defined as constraints under which the system must operate and the standard which must be met in the delivered system.

#### User Interface

- The system is required to have an interface that is specifically intended to allow the user to access the internal components of the system in a relatively easy fashion and without having to know specifically how everything is put together or how they work together. A common interface must be created in order to reduce the learning curve of the users. For this system, the user is just using all the common creation to access location without using keyboard. The mouse will become the main input for this system.

#### Reliability

- The system should be reliable which mean it does not produce dangerous or costly failures when it is used in a reasonable manner. That is in a manner that a typical user expects as normal. This definition recognize that a system may not always be use in the way the designers expect.

#### Efficiency

- A process or a procedure that can be called or accessed in an unlimited number of times to produce the similar outcomes or output with predictable pace or speed.

#### Simplicity

- A form or screen properly uncluttered in a manner that focuses the user attention and its basic layout doesn't confuses the user.

#### Understandability

- In term of recovering method used, allows other programmers to understand the logic of program flows thus changes made easily upon the necessary program segment without modifying essential part of the program. Simple and clear sentences or messages are displayed so that user can use the system without difficulty.



**Modularity**

- Modularity is a key factor in good program design. The working of the system was broke into modules so that the distinct functions of objects could be isolated from one another. This characteristic makes it easier to perform testing and maintenance.

**Maintainability**

- Maintainability can be defined as the ease with which software can be understood , corrected , adapted and/or enhanced in the future.

**Help**

- The system should provide help to guide and clarify to the user about the usage of the system. In this system, there will be instructions for the user to tell them what to do next. Description of certain buttons will be given to let the users know the function of the button.

**3.5.4 Prototyping Requirement**

Prototyping means develop a system based on the users criteria to see whatever the users criteria could be fulfilled . Meanwhile a part of the system is provided to let users evaluate to improve the system. This may mean that users could check whatever they are satisfied with all the requirement prototyping takes to form. There are two prototypes that can be used ,one is the throwaway prototyping and the evolutionary prototype .In this teaching package an evolutionary prototype was developed.

**3.5.5 Runtime Requirements****Hardware requirements:**

- Intel Pentium II 166 MHz or better
- Windows 95, 98, 2000, Windows NT 4
- Min 32 MB RAM
- Color Monitor

- CD-ROM drive
- 4MB SVGA graphics adapter
- CD Writer
- Keyboard, Mouse
- Speaker, Microphone
- 16 bit sound card
- 10 GB of available disk space

### Software Requirements

- Windows 95/98/2000
- Internet Explorer 5.0
- Macromedia Director 8.0
- Adobe PhotoShop 6.0
- Flash 5.0
- Microsoft Access 2000
- Sound Forge 5.0



### Chapter Summary

This chapter includes the system analysis that was done. It also contains the methodology that is used in creating this multimedia teaching package which is the modified version combining the benefits of both waterfall model and prototyping approach. Here is where the chosen tools are explained. All the requirements of the system including the non-functional, functional and prototyping requirements are explored in this chapter.

**List Of Reference**

- 1) Kenneth E. Kendall and Julie E. Kendall, System Analysis and Design, 3<sup>rd</sup> edition , Prentice Hall International:1996
- 2) Shari Lawrence Pfleeger, “Software Engineering, Theory and Practice”, Prentice Hall: 1998
- 3) Sellapan, P., Software Engineering : Managements and Methods, 1<sup>st</sup> Edition, Sejana Publishing 2000.
- 4) <http://www.macromedia.com>
- 5) <http://www.adobe.com>
- 6) [http:// soundfourge .com](http://soundfourge.com)



## CHAPTER 4 - SYSTEM DESIGN

### 4.1 System Design

System design is a very essential in a development process as it plays a major role in determining the success of the system. The system specifications describe the features of a system and their arrangements to users. The requirements that are provided from the analysis phase are translated into design specifications. Integrate the interface design into the overall design. The interface should complement the whole package, and reflected in the end, factors such as navigation and interactivity are often interface problems other than anything else. In this chapter the specification of each chapter will be given. After doing a thorough analysis, I can now design a system that will meet the requirements. The design process is a continuous one, and the design must be refined and improved as the system evolves.

# CHAPTER 4 SYSTEM DESIGN

### 4.2.1 Introduction

An introduction on XML, its history and its use. The use of XML is a very popular and what advantages it has compared to SGML and HTML. It will explain the difference between XML and HTML.

### 4.2.2 Lessons

This module has 10 Chapters. Below the chapters are explained briefly on what does it overall cover on.

#### Chapter 1-5.xml

This chapter explains about the concept of a XML document. How it is declared and how does a XML document look like.

## CHAPTER 4 – SYSTEM DESIGN

### 4.1 System Design

System design is a very essential in s development process as it plays a major role in determining the success of the system. The system specifications describe the features of a system and their appearance to users. The requirements that are concluded from the analysis phase are translated into design specifications. Integrate the interface design into the overall design. The interface should compliment the whole package , not be tacked on at the end. Issues such as navigation and interactivity are often interface problems rather than anything else. In this chapter the specification of each chapter will be explained. After doing a through analysis, I can now design a system that will meet the requirements. The objective of this phase is to design an effective system that meets the requirements and is within the constraints bounds. This is just a basic logical design that is going to be used in designing Learning XML and it is subjected to changes.

### 4.2 Overview of Learning XML Modules

#### 4.2.1 Introduction

- An introduction on XML. It contains the history and the main usage of XML. It explains why XML is becoming more popular and what advantages it has compared to SGML and HTML. It will explain on what is SGML and HTML.

#### 4.2.2 Lessons

This module has 10 Chapters. Below the chapters are explained briefly on what does it overalls contain.

##### Chapter 1-Syntax

This chapter explains about the syntax of a XML document. How it is declared and how does a XML document has a valid syntax.



## Chapter 2- Elements

This chapter explains on the important elements in a XML document.

It shows what does an element has and how is it declared and the content model.

## Chapter 3- Attributes

This chapter explains on the attributes of a XML document and how does it differ from HTML. The types of attributes , the values and constraints, the default specification and the multiple attributes declaration for an element is also explained.

## Chapter 4-Writing a File

This chapter teaches users to write a XML file. It explains the procedure on writing the file. Such as how to create the file ,and the next step is to write the declarations. Followed by adding a comment, defining the root element, and etc. There are many more procedures which will be explained in the teaching package

## Chapter 5 –The XML DTD

This chapter explains on what is a DTD , why is it used. It also tells how to get stated on a DTD and the types of XML document content. It tells how is the anatomy of a DTD and also the syntax of a DTD.

## Chapter 6-Validation on Schemas

This chapter explains on what is the correct syntax and schemas for a Well Formed XML .It tells which is a valid XML and it is validated against DTD.

## Chapter 7 – XML Parsers

This chapter explains on the different XML parsers. To read and update, create and manipulate an XML document, a parser is needed. The parsers that will be thought on how to use is a parser and how to load an XML file into the parser.

## Chapter 8- XML and other Specification.

This chapter explains about the high level overview of various XML-related specification and what they mean. There is a lot going on around XML, so there is a lot to learn. This chapter will help users to refer to it whenever they came across something they are reading. At least it will users to be at least “conversant “ with XML.

### **4.2.3 Tutorials**

There are 2 situational questions where it summaries what have been thought in the chapters into 2 different situation. Situation 1 covers what have been thought from Lesson 1 to Lesson 4 that is to write a well-defined XML file. Situation 2 is about Lesson 5 that covers about the DTD.

### **4.2.4 Library**

A repository on XML coding and other important facts. This is to help users to get ready codes and tags so it will be easier for them to use the create their own XML documents.

### **4.2.5 Appendix**

The Appendix contains important keywords in XML and the teaching package. It contains all the important keywords in the package so users can find, the terminology or explanation they are looking for.



### 4.3 System Modeling

A system is a representation of an in-place or proposed system that describes the data flow through out the structure. The model describes the point where data or information enters a system and the places where it will be processed, as well as the actions taken and the points where data will be output.

A system model is documented through a variety of design diagrams. A design diagram is a graphic or visual representation of a structure. Design diagrams include data flow diagram (DFD), structured charts, decision trees, and other items.

The DFD is used as the system-modeling tool for this project so as the structure chart.

A DFD is a graphical illustration that shows the flow of data and logic within a system.

Reasons why DFD modeling system is used:

- It gives me the freedom from committing to the technical implementation of the system too early. This is because the symbols used does not specify the physical aspects of implementation. Therefore it still gives me room for changes and enhancing.
- Using DFD also gives me a better understanding of the interrelatedness of systems and subsystems.
- It allows me to differentiate the system from its environment by locating its boundaries.

- This model also allows me to communicate the current system knowledge to the lecturer, whereby DFD shows an incomplete representation of my understanding, therefore any required changes can be incorporated to the system effectively. Necessary background is provided so that the data flow diagram will be meaningful rather than confusing.
- This model also allows me describe each component used in the diagram whereby to ensure all necessary output can be obtained from the input data and that processing logic is reflected in the diagram. Moreover it also allows early detection and correcting errors and design flaws in the early stage.



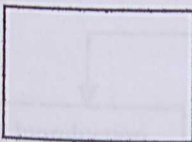
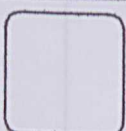

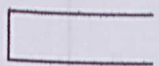

<u>Symbol</u>	<u>Name</u>	Description
	Source or destination of data.	External sources or destination of data, interact with the system but are outside its boundary.
	Process	It represents the transformation or processing of information within a system.
	Selection	It is used to make a choice that interacts with the system.
	Data store	It is used for showing the data storage or referenced by a process.
	Data flow	It is used to show the movement of data from an origin to a destination with the head of arrow pointing towards the destination.

Figure 4.1 Representing the structured module of Learning XML

4.4 Structured module of Learning XML

Representing the structured module of Learning XML.

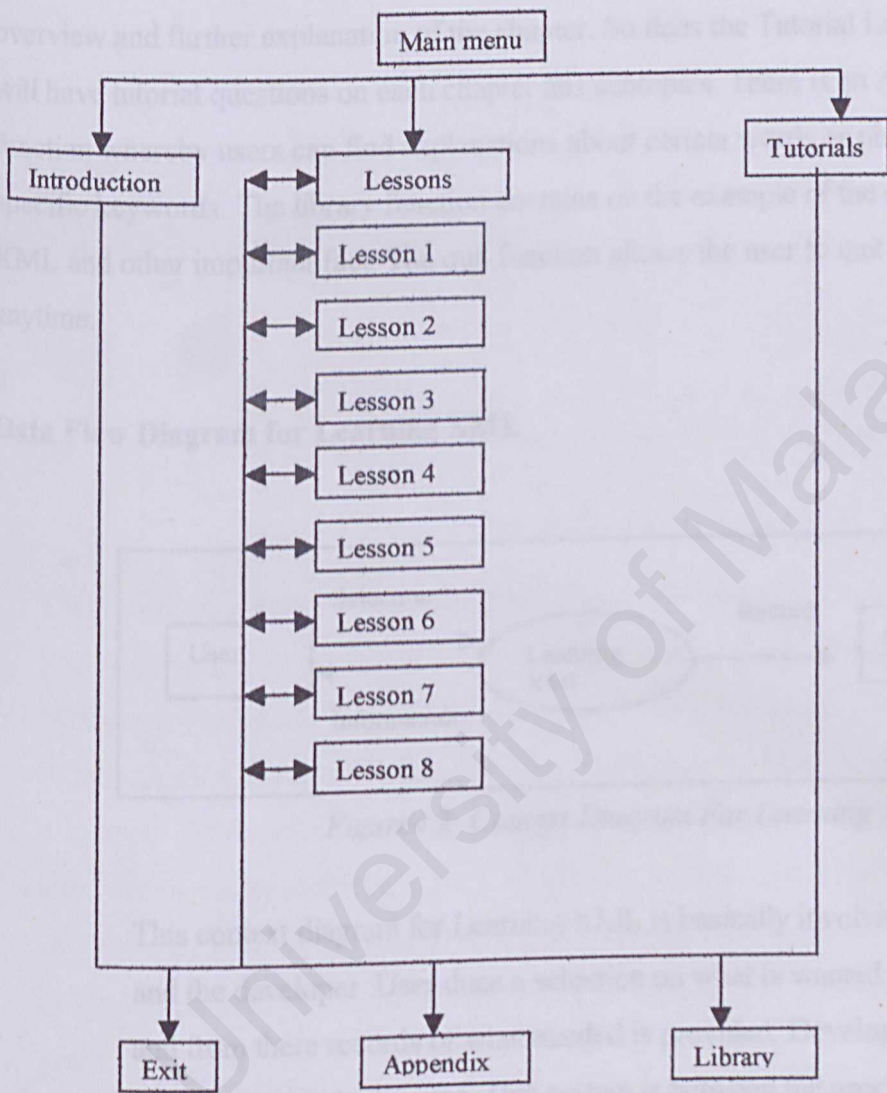


Figure 4.1:Representing the structured module of Learning XML



Figure 4.1 is a concept map that illustrates the structure of the Learning XML module. Learning XML will be divided to three levels: - Introduction level, Lessons level and Tutorial level. In the Lesson level it is broken up to 8 chapters whereby each chapter is divided into several subtopics. Each subtopic will give the user an overview and further explanation of the chapter. So does the Tutorial Level, which will have tutorial questions on each chapter and subtopics. There is an Appendix function whereby users can find explanations about certain words or phrases using specific keywords. The library function contains on the example of the codes on XML and other important fact. The quit function allows the user to quit the program anytime.

#### 4.5 Data Flow Diagram for Learning XML

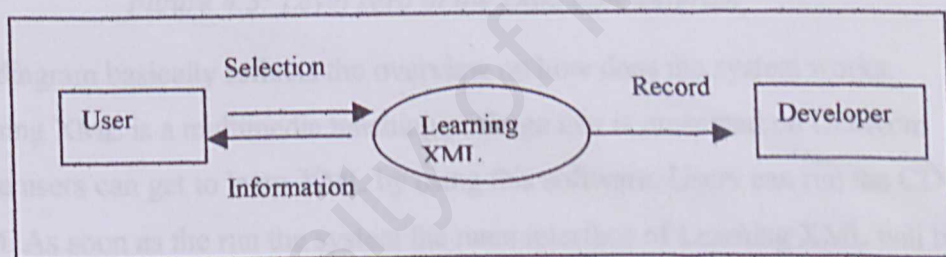


Figure 4.2: Context Diagram For Learning XML

This context diagram for Learning XML is basically involving the users and the developer. User does a selection on what is wanted to the system and from there records of what needed is provided. Developer sends back the information to the user. This system is between the user, the teaching package and the developer.

**Level Zero Of Data Flow Diagram**

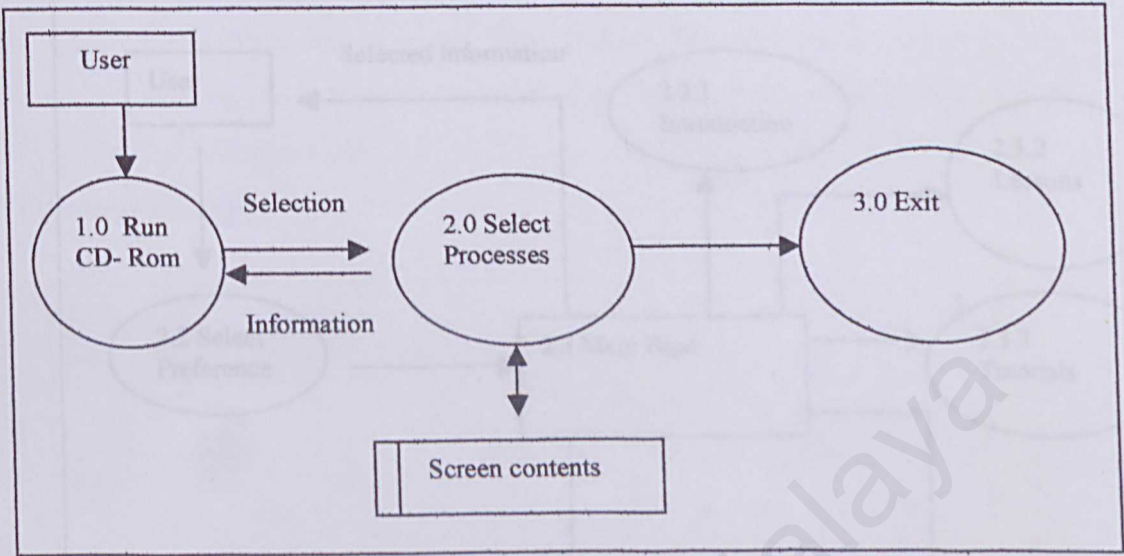


Figure 4.3: Level zero of the Data flow diagram

This diagram basically reflects the overview on how does the system works. Learning XML is a multimedia teaching package that is presented on CD-Rom where users can get to learn XML by using this software. Users can run the CD-ROM. As soon as the run the system the main interface of Learning XML will be displayed. There users can select on what do they want to access. The screen information will be displayed after the selection is done. Users can do as many selections by clicking the back button or the home button in case of the need to go to another module. Users can exit the system after the information needed is given.



**Level One Of the Data Flow Diagram**

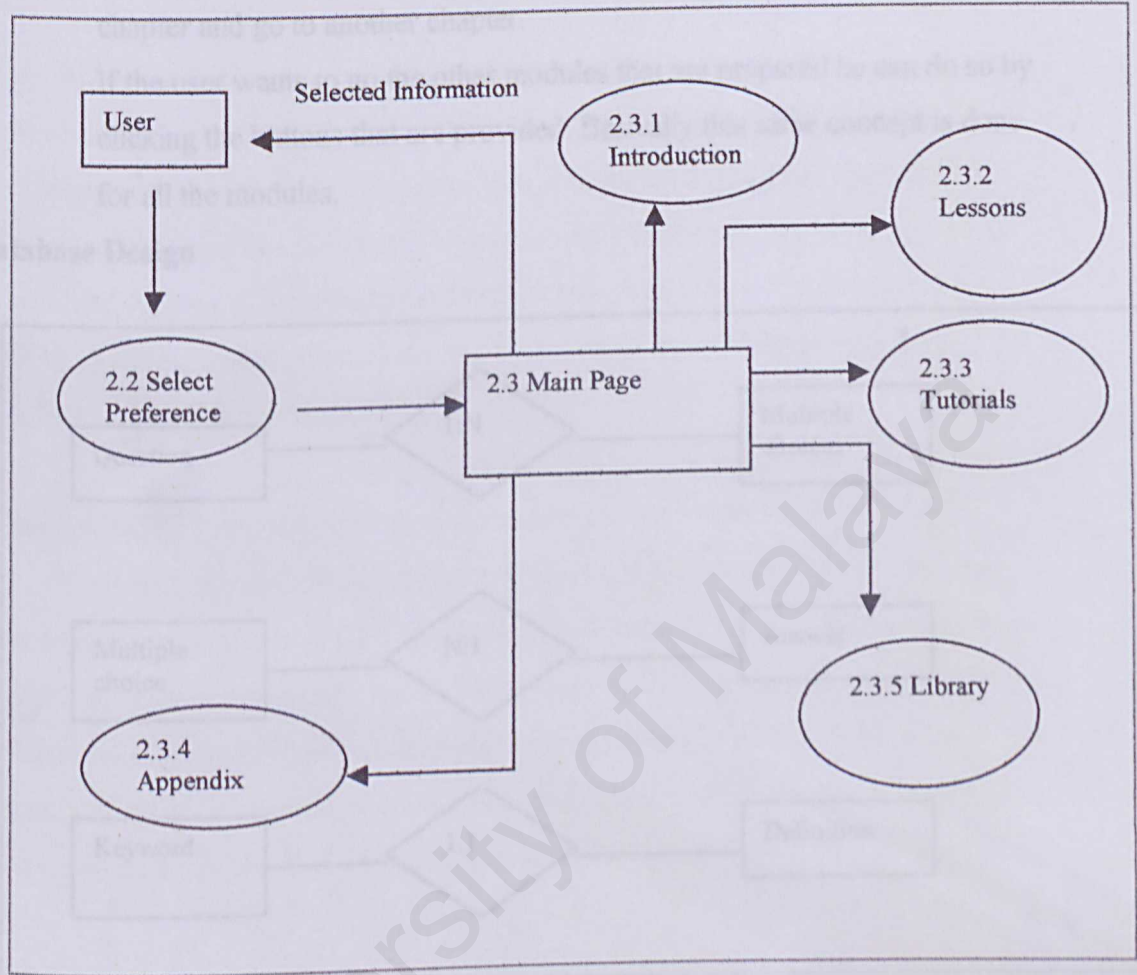


Figure 4.4: Level One of DFD For Select Processes

This data flow diagram actually explains on the level one of the Select Processes that was done in Level Zero. It shows that as the user comes to the process of Select Preference process. From here the main menu of Learning XML is displayed. Here the user can make a selection on what module does it want to go. Users can select Lessons module and go through the chapters that are provided there. From there it can go back to the main menu and exit after that or go the other modules. It goes for the other modules, selection can be done and then user can go through the modules that are prepared and later exit the system. As the user is going

through a chapter in the lesson module it can choose to skip the next chapter and go to another chapter.

If the user wants to go the other modules that are prepared he can do so by clicking the buttons that are provided. Basically this same concept is done for all the modules.

#### 4.6 Database Design

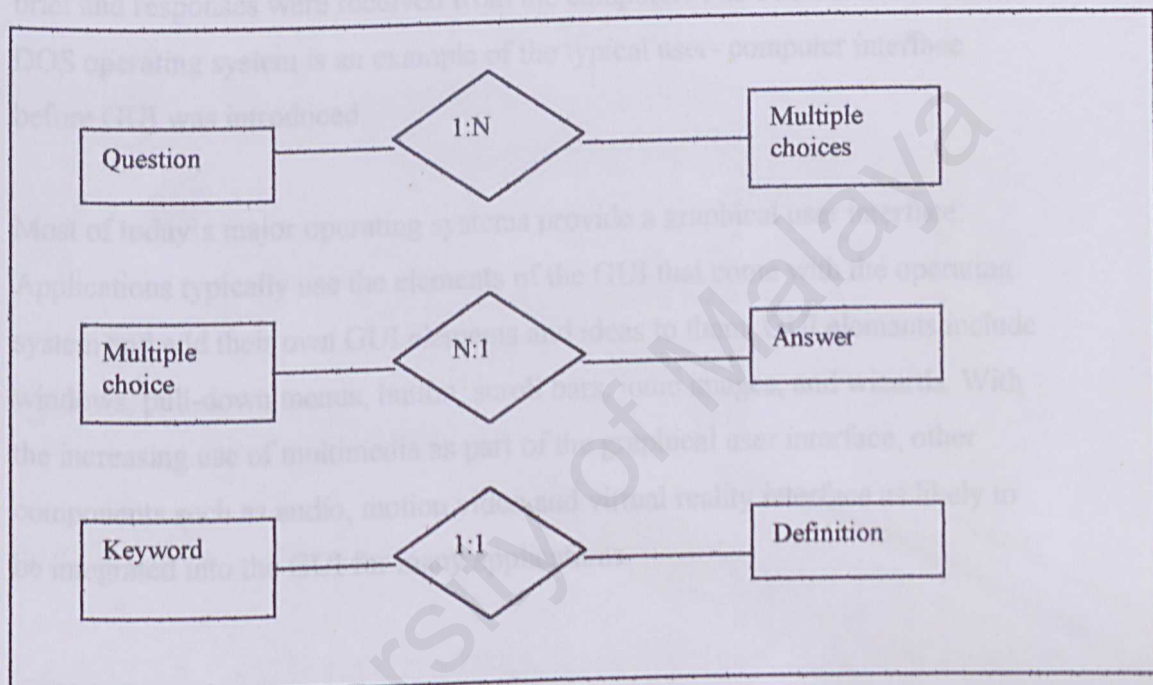


Figure 4.5: Basic Database Design

The database is used to store tutorial questions and answer and also the quiz Q&A. all the questions are in multiple choice answer there fore the user will just have to click the answer. Each question has four multiple choice answer and for a set of multiple choice answer there will be just one answer. This relative relational will be shown in Ms Access which will be used as the software to store the data. The database is also used to store the definition of the keywords from the Index function. The entity is 1:1 due to the fact that for every specific keyword there.



## 4.7 Graphical User Interface

A GUI is a graphical rather than a purely textual user interface for communication between the user and a computer. The term GUI came into existence because the first interactive user interfaces to computer was text- and-keyboard oriented and did not include graphics. Commands had to be typed in brief and responses were received from the computer. The command interface of DOS operating system is an example of the typical user- computer interface before GUI was introduced.

Most of today's major operating systems provide a graphical user interface. Applications typically use the elements of the GUI that come with the operating system and add their own GUI elements and ideas to them. GUI elements include windows, pull-down menus, button, scroll bars, ionic images, and wizards. With the increasing use of multimedia as part of the graphical user interface, other components such as audio, motion video and virtual reality interface as likely to be integrated into the GUI for many applications.

4.8 User Interfaces

Below are the layouts of the interface design for ITFA. It is not a permanent design and can be subjected to changes. This is just to show how the main pages will look in the system when its s designed.

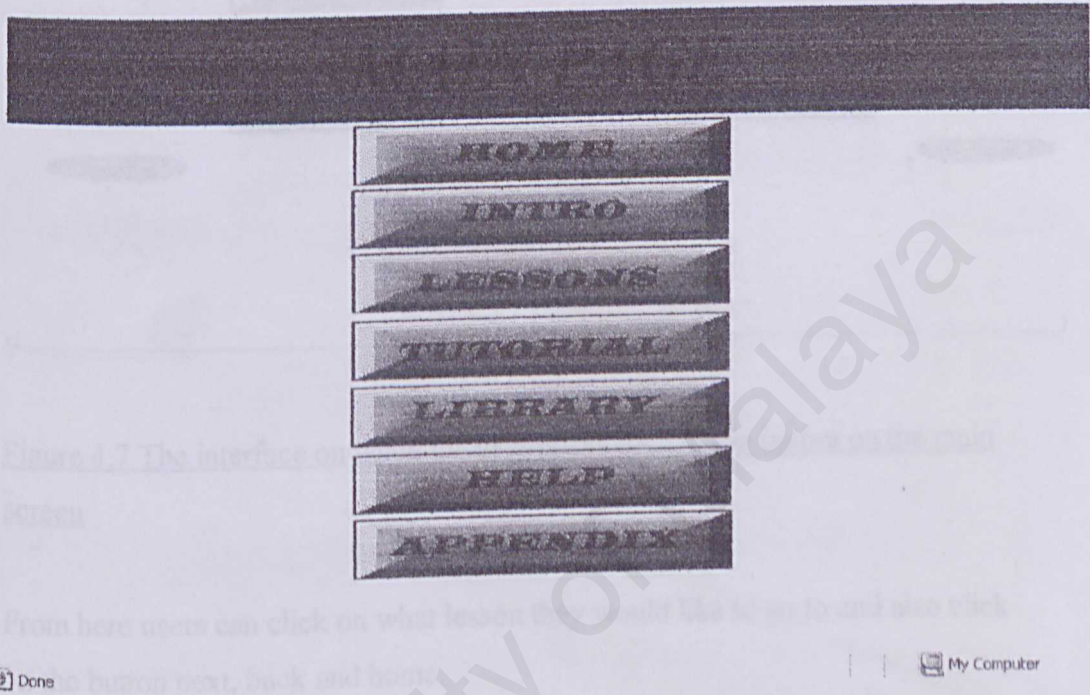


Figure 4.6 The example of the interface when the user comes to do selection on what it wants from the system



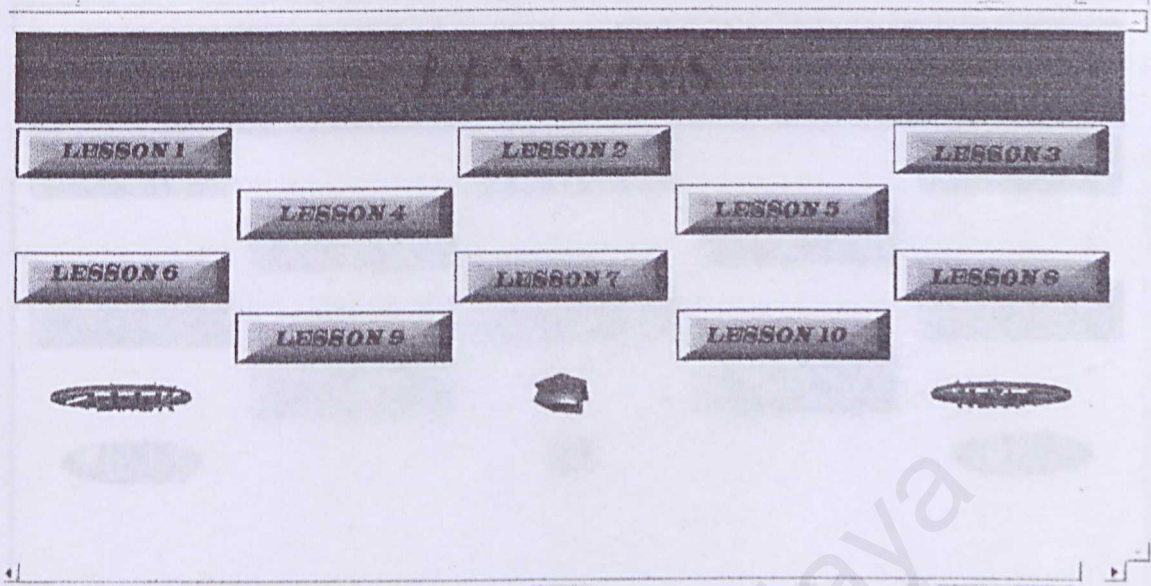


Figure 4.7 The interface on when the user clicks the Lesson button on the main screen

From here users can click on what lesson they would like to go to and also click on the button next, back and home.

#### Chapter Summary

This chapter is about the design of the multimedia teaching package. It describes the typical design of this system. It also describes the design of the system in terms of the user interface. It is that chapter that describes the design of the system. It explains about the design of the system.

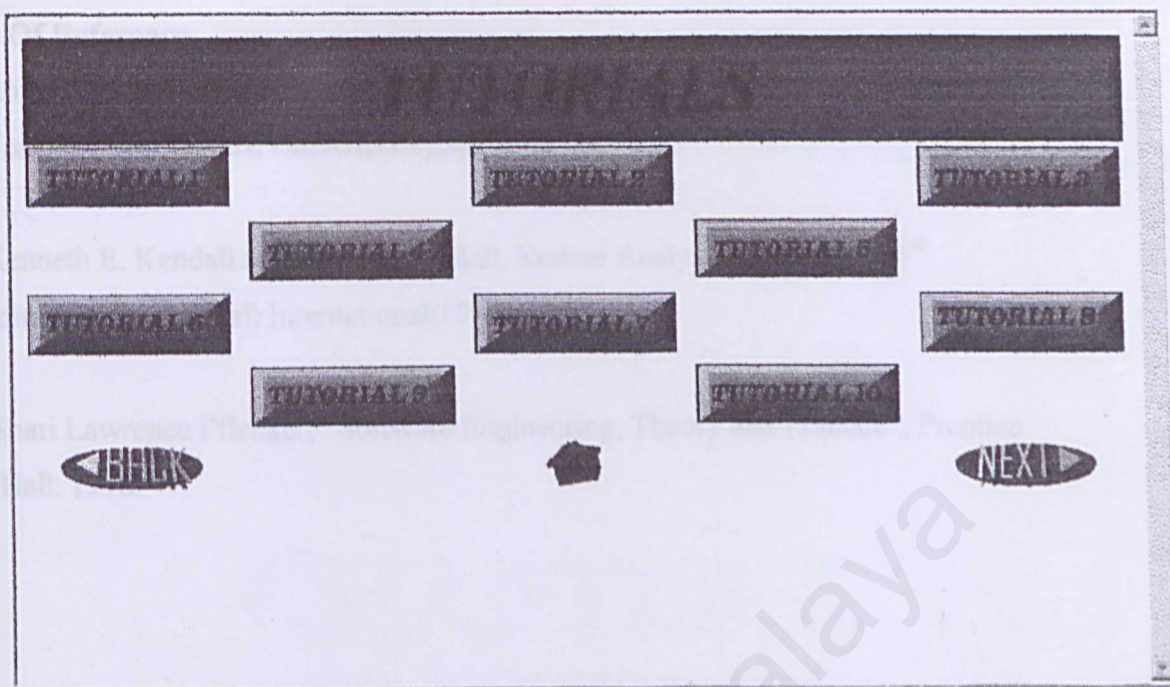


Figure 4.8 The interface for the selection on tutorials.

It basically works in the way as the lesson interface and the buttons work.

### **Chapter Summary**

This chapter is about the system design. This chapter describes the logical design of this system. It includes the systems modules design on what is going to thought in that chapter, system data flow design and user interface design. It explains about the Graphical User Interface.



### List Of Reference

- 1) Ali, Benforoz and Hudson , Frederick J., Software Engineering Fundamentals, New York, Oxford, Oxford University Press; 1996.
- 2) Kenneth E. Kendall and Julie E. Kendall, System Analysis and Design, 3<sup>rd</sup> Edition , Prentice Hall International:1996
- 3) Shari Lawrence Pfleeger, “Software Engineering, Theory and Practice”, Prentice Hall: 1998.

## CHAPTER 5: SYSTEM DEVELOPMENT AND IMPLEMENTATION

### 5.1 Introduction

This section will discuss the development strategy as well as the implementation for the whole package. System Development is one of the critical stages in developing a system. System Implementation is a process that converts the system requirements and design into programming codes. This phase is where the system is being modified as to the user's design.

# CHAPTER 5 SYSTEM DEVELOPMENT AND IMPLEMENTATION

- Project Planning (1)
- 34 XCD (2)
- 125 M (3)
- 20 (4)
- 1 (5)
- 1 (6)
- 1 (7)
- 1 (8)
- 1 (9)
- 1 (10)
- 1 (11)
- 1 (12)
- 1 (13)
- 1 (14)
- 1 (15)
- 1 (16)
- 1 (17)
- 1 (18)
- 1 (19)
- 1 (20)
- 1 (21)
- 1 (22)
- 1 (23)
- 1 (24)
- 1 (25)
- 1 (26)
- 1 (27)
- 1 (28)
- 1 (29)
- 1 (30)
- 1 (31)
- 1 (32)
- 1 (33)
- 1 (34)
- 1 (35)
- 1 (36)
- 1 (37)
- 1 (38)
- 1 (39)
- 1 (40)
- 1 (41)
- 1 (42)
- 1 (43)
- 1 (44)
- 1 (45)
- 1 (46)
- 1 (47)
- 1 (48)
- 1 (49)
- 1 (50)
- 1 (51)
- 1 (52)
- 1 (53)
- 1 (54)
- 1 (55)
- 1 (56)
- 1 (57)
- 1 (58)
- 1 (59)
- 1 (60)
- 1 (61)
- 1 (62)
- 1 (63)
- 1 (64)
- 1 (65)
- 1 (66)
- 1 (67)
- 1 (68)
- 1 (69)
- 1 (70)
- 1 (71)
- 1 (72)
- 1 (73)
- 1 (74)
- 1 (75)
- 1 (76)
- 1 (77)
- 1 (78)
- 1 (79)
- 1 (80)
- 1 (81)
- 1 (82)
- 1 (83)
- 1 (84)
- 1 (85)
- 1 (86)
- 1 (87)
- 1 (88)
- 1 (89)
- 1 (90)
- 1 (91)
- 1 (92)
- 1 (93)
- 1 (94)
- 1 (95)
- 1 (96)
- 1 (97)
- 1 (98)
- 1 (99)
- 1 (100)

### 5.2 The software required during testing is:

- Operating System - Microsoft Windows 98
- Director 8.0
- Adobe Photoshop 6.0
- Flash 5.0
- Sound Forge and Cool Edit 2000



## CHAPTER 5: SYSTEM DEVELOPMENT AND IMPLEMENTATION

### 5.1 Introduction.

This section will discuss the development strategy as well as type of implementations for the whole package. System Development is one of the critical stages in developing a system. System implementation is a process that converts the system requirements and design into programming codes. This phase at some point involves some modifications to the previous design.

### 5.2 Development Environment

Development environment is of vital importance to any system development. It consists of hardware and software requirements. By using the suitable hardware and software, the process of development able to speed up.

#### 5.2.1. Hardware Tool

- Pentium Pentium III 266 Mhz Processor
- 34 X CD Rom Drive
- 128 Mhz RAM
- 20 GB Hard disk
- Microphone
- Other standard desktop PC component

#### 5.2.2 The software required during testing is:

- Operating System – Microsoft Windows 98
- Director 8.0
- Adobe Photoshop 6.0
- Flash 5.0
- Sound Forge and Cool Edit 2000

Since this package is about multimedia learning, prototyping should be done. Below are reasons why prototyping is needed:

- Easily change the system should any problem encountered.
- User interface used could well reflect the user needs.
- Able to find any missing functions and requirements.
- Since this package has already existed in the market, prototyping could be used as a template for other implementations.

### 5.3 Development Of the System

For my multimedia package system, the system development tool that was used is Director 8.0 and the coding or programming language that involves divided to two. They are XML (Extensible Markup Language) and Lingo scripts. The system was developed using a modular approach which means that every function that is written in the development could be called by other functions in the system. With this all the multimedia elements such as text ,audio , video , graphics and animation could be integrated in the system to make it more interesting.

#### 5.3.1 A Brief Introduction about Director 8.0

Director is an essential piece of multimedia *authoring tool*, allowing you to create your own interactive multimedia software packages to run on Mac or PC even if user has broad programming experience. Director is actually frame- based animation program. Someone who applied animation cell management to a computer spread sheet originally designed .Each cell of the spread sheet contained graphics into in the form of a bitmap. Then the spread sheet becomes a 'score' .When the score's cell was automatically sequenced using the computers built- in –clock , the bitmap images would sequence on the screen in a window. Animation also can be inserted which will be controlled by a spread sheet. Director is the main tool used for authoring CD-ROMs . It was originally designed as a package to integrate black and white animation and sound, Director has evolved into a sophisticated tool allowing you to combine text, sound, animation, graphics and video, and to add user interaction using the high-level Lingo language.



Simple diagrams and texts , as well as sound can be supplementary product line – produced with Macromedia Director. However the used data or actor are imported by special programs .

The user surface of the Macromedia Director is divided into 4 main parts :

- The stage –the backdrop on which all elements of the multimedia presentation will perform .The elements are named cast in Macromedia Director. This is where all the objects of the application will be displayed.
- Occupation Place where raw media or also data which are involved in the project.
- The control desk -Control of trial runs with the creation of the project.
- The film Script-This is where the script of the film is listed which actor when where like for a long time in the project occurs.

Each module was built with each cast interaction with each other. To first start the development of the system , 1<sup>st</sup> each cast is imported from the files they are needed from and automatically they will be placed as an Internal Cast. This cast are in the form of text, audio, video, graphic or animation .From there this Internal Cast will be dragged to the Score window and will be arranged accordingly on the Stage. Score is where all interaction in a module takes place. As shown in the example below the cast members for the main page of Learning XML are paced accordingly and the duration on which it has to take part in a movie is placed in the Score.

The Property Inspector tells the behavior of each cast.

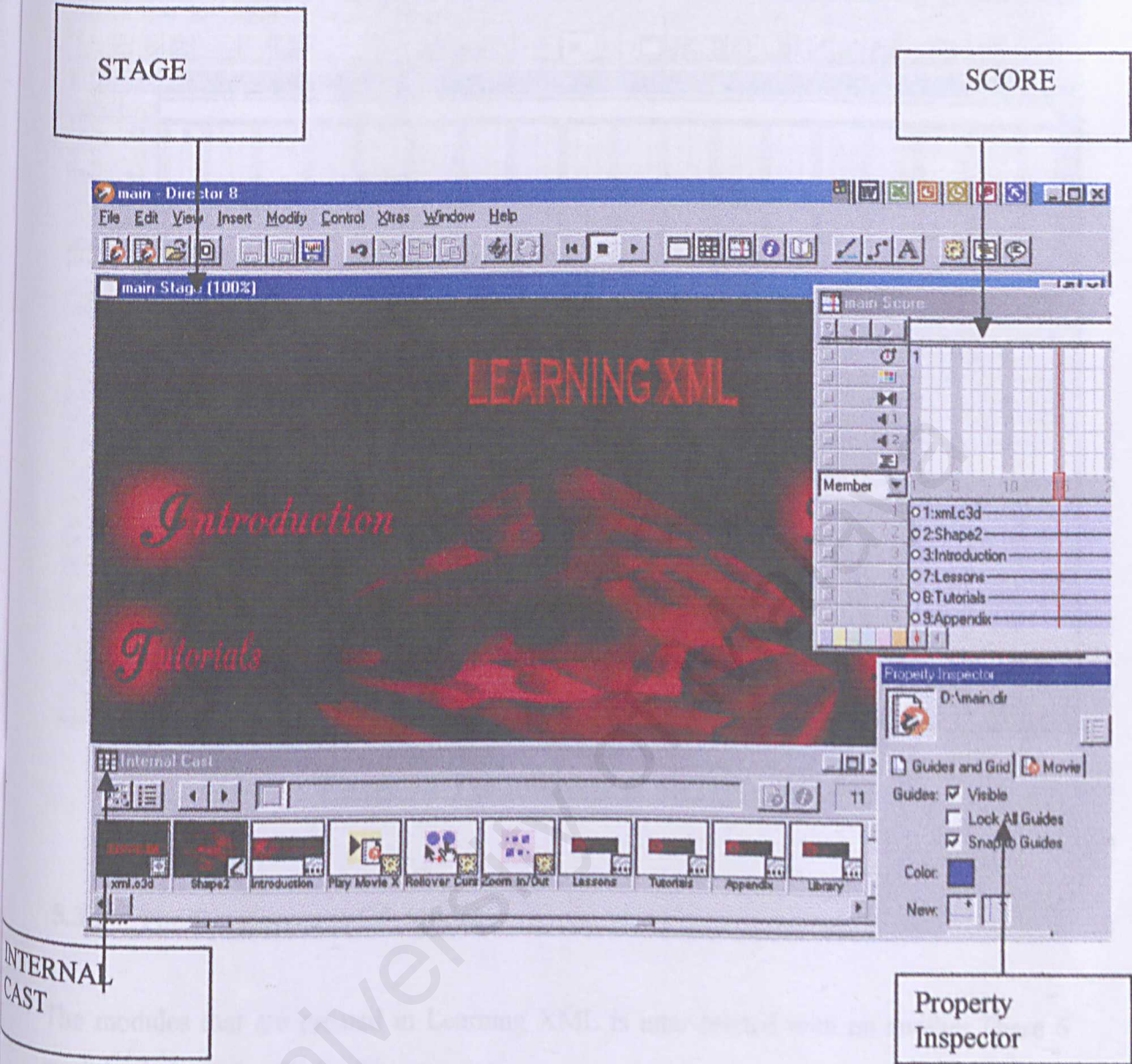
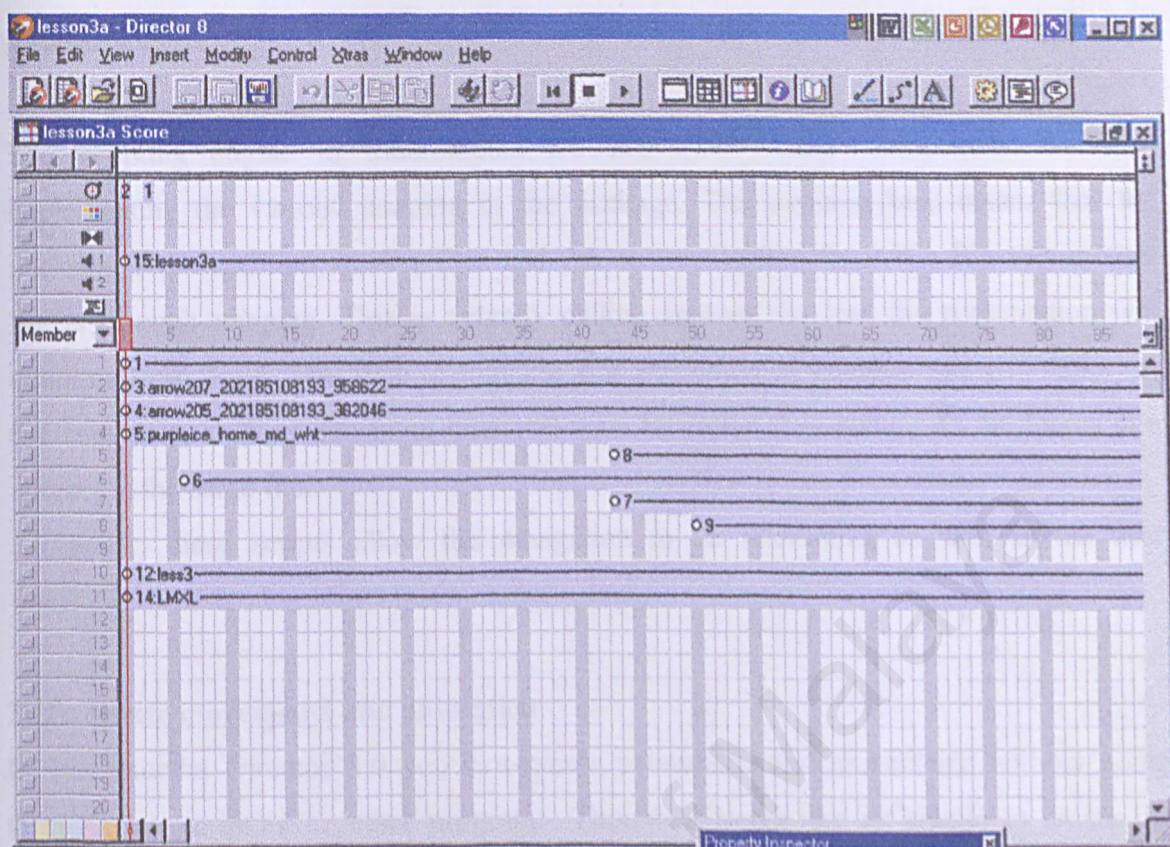


Figure 5.1 The main page –an interface in Macromedia Director 8.0





**Figure 5.2 The Interface of the Score Window**

### 5.3.2 The Development Of Modules

The modules that are created in Learning XML is inter-related with on another. There 6 modules in Learning XML. They are:

- 1) Introduction
- 2) Lessons
- 3) Tutorials
- 4) Library
- 5) Appendix

To develop each nodule in this teaching package , there were certain things that had to be considered. Such as:



### 1) Lingo Scripts

The scripting engine in Macromedia Director 8.0 – Lingo Scripts, is useful in implementing the various jumping in the package. The jumping here means that the package is divided into smaller files and the communication from one file and another file is through jumping. This is actually calling another file and the presentation will continue from there. This type of jumping was implemented because this will make the file smaller as well as execution time will be a lot faster than putting all the topics in a subject in the same file.

Moreover, these scripts allow the navigation to be interactive as well as the user interface.

### 2) To create movements and animation

In Macromedia Director 8.0 you can create animations and movements without going through any scripting or other programs. Though it looks like its simple to create the animation, but it is very tedious. You have to make sure each movement is right and it takes a long time just to get one movement done. Especially where there are many cast members involved. Things can get pretty mess up. To make sure the creation of animation and movements can be done smoothly, cast objects should be sorted.

To create a smooth animation many cast members have to take part so sorting is done so the important cast will be given priority. The speed of the movement must be controlled using 'tempo' and also to handle the 'score'.

### 3) Inserting Audio (Sound)

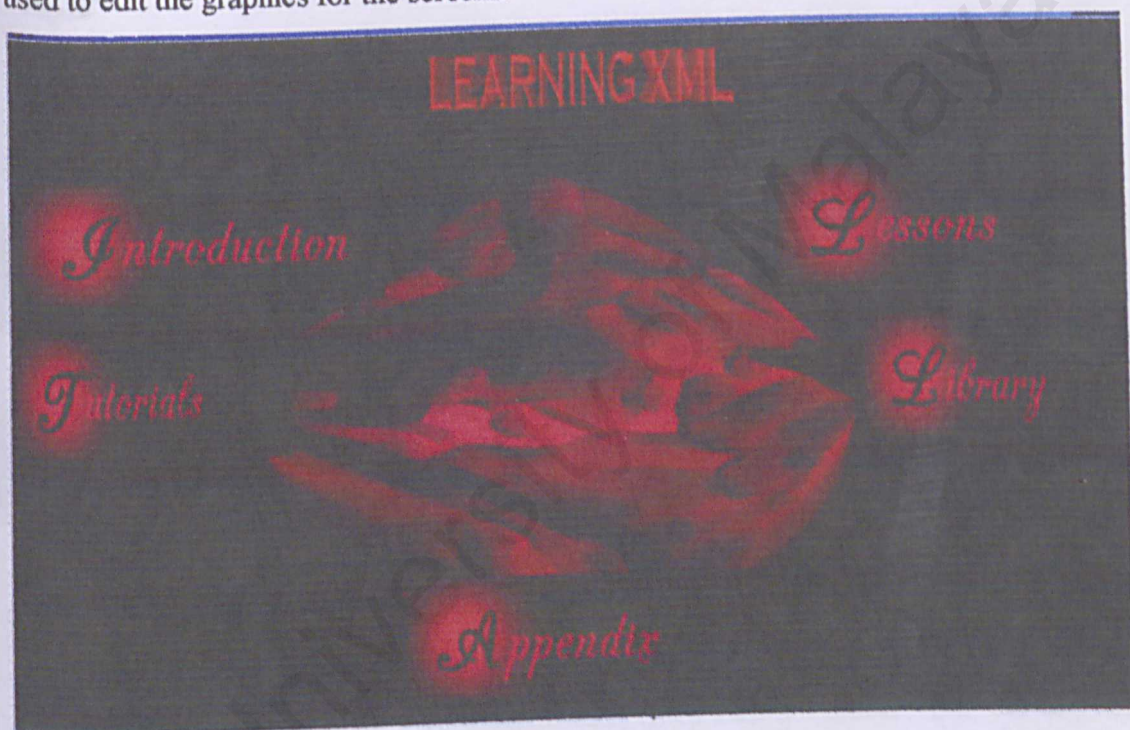
Macromedia Director 8.0 supports 2 formats of sound which are **\*.AIFF (Audio Interchange File Format)** and **\*.WAV (Wave form Audio File)**. The basic way to insert audio files are as the same as getting a cast member that is to import. The only difference is that the sound file is placed at the 'effect section' on the score. It cant be placed on the stage like other cast members. The sound editor that I used, CoolEdit 2000 was in the form of \*.wav file.



### 5.3.3 The Development of the Screen Interfaces

The screens in this teaching package were developed using different colors to make it attractive. The colors chosen are very light colors so that it doesn't cause a pain to the eye. It is rather simple because this package is a text-based package where there are more text involved compared to pictures. To develop each screen, the font, background color and colors on the text was done with a lot of consideration.

Each module in this teaching package is different. This is to show the difference in the modules so users can differentiate in what module they are. Adobe Photoshop 6.0 was used to edit the graphics for the screens.



**Figure 5.4 The Interface Of The Main Screen In Learning XML**

### 5.3.4 Buttons and icons

In this multimedia teaching package there only a few buttons were used to navigate through the system. This buttons are the standard buttons that are used which is basic and easy to understand. When a user puts the mouse over a button or icon the mouse will change to a 'closed hand' mouse that is used to make it attractive.

Below the example of icons and buttons that used in the teaching package is shown:



This button is used in each screen to take users back to the main page.



In some of the screen this color of the home buttons is used .



This arrow is used to show the function of previous which was used in the Lesson module and also the Appendix module



This arrow is used to show the function of next which was used in the lesson module and also the Appendix module



This arrow is used to show the previous function in the Tutorials module and the Library module



This arrow is used to show the previous function in the Tutorials module and the Library module



## 5.4 Coding

### 5.4.1 Extensible Markup Language (XML)

This language is done basically to teach, as it is the title of my multimedia-training package. There are not so much difficulty in presenting and learn this language although it is still new. This is because the syntax and semantic of the language are similar to other languages that I have learned such as HTML.

### 5.4.2 Lingo Scripting

The scripting engine in Macromedia Director 8.0 – Lingo Scripts, is useful in implementing the various jumping in the package. The jumping here means that the package is divided into smaller files and the communication from one file and another file is through jumping. This is actually calling another file and the presentation will continue from there. This type of jumping was implemented because this will make the file smaller as well as execution time will be a lot faster than putting all the topics in a subject in the same file.

Moreover, these scripts allow the navigation to be interactive as well as the user interface.

Example of a Lingo Script that was used throughout the multimedia media teaching package. This script is called 'Rollover Cursor Change' because when the mouse is used on it the mouse changes to the image of a closed hand.

-- DESCRIPTION --

```
on getBehaviorDescription me
  return \
    "ROLLOVER CURSOR CHANGE" & RETURN & RETURN & \
    "Changes the cursor when the mouse rolls over the
current sprite." & \
    "Choose one of the pointers included with Director, or
specify two 1-bit 16x16 pixel bitmap members: one to act as
the pointer image, the other to define the
transparent/opaque areas of the cursor." & RETURN & RETURN
& \
  "TIPS:" & RETURN & \
```

```

    "Place a single pixel at the topRight and bottomLeft of
    the image itself to create what is in fact a 17x17 pixel
    bitmap." & \
    "These extra pixels will not appear in the cursor (they
    will be clipped) but the mask will align with them." & \
    "This ensures that the opaque area surrounds the cursor
    image correctly." & RETURN & RETURN & \
    "Set the regPoint of the image to define the cursor's
    hotspot." & RETURN & RETURN & \
    "PARAMETERS:" & RETURN & \
    "* EITHER - Use one of Director's built-in cursors." &
RETURN & RETURN & \
    "* OR - Use your own bitmap images." & RETURN & \
    "* Custom Image " & RETURN & \
    "* Custom Mask" & RETURN & RETURN & \
    "To use custom images, ensure that " & QUOTE & "1-bit
    bitmap" & QUOTE & " is selected as the type of cursor."
end getBehaviorDescription

case myCursorType of
on getBehaviorTooltip me
    return \
        "Use with graphic members." & RETURN & RETURN & \
        "Modifies the cursor when the mouse rolls over a
    sprite." & RETURN & RETURN & \
        "You can use built-in or custom images for the cursor."
end getBehaviorTooltip

-- PROPERTIES --

property spriteNum
-- author-defined parameters
property myCursorType
property myBuiltInCursor
property myCursorMember
property myCustomCursor
property myCustomMask
-- internal properties
property mySprite
property mySavedCursor

-- EVENT HANDLERS --

on beginSprite me
    SetSpriteCursor me
end beginSprite

on endSprite me

```



```

    mySprite.cursor = mySavedCursor
end endSprite

-- CUSTOM HANDLER --
on SetSpriteCursor me

    mySprite = sprite (me.spriteNum)
    -- Save cursor to revert to
    mySavedCursor = mySprite.cursor

    -- Set the cursor of the sprite
    if voidP (myCursorType) then
        mySprite.cursor = myBuiltInCursor
        exit
    end if

    case myCursorType of
        "Built-in cursor":
            mySprite.cursor = myBuiltInCursor
        "Cursor Member":
            myCursorMember = value (myCursorMember)
            cursorList = [myCursorMember.number]
            mySprite.cursor = cursorList
        "1 bit bitmap":
            myCustomCursor = value (myCustomCursor)
            cursorList = [myCustomCursor.number]
            if myCustomMask <> "no mask" then
                myCustomMask = value (myCustomMask)
                cursorList.append(myCustomMask.number)
            end if
            mySprite.cursor = cursorList
    end case
end SetSpriteCursor

-- AUTHOR-DEFINED PARAMETERS --

on isOKToAttach (me, aSpriteType, aSpriteNum)
    case aSpriteType of
        #graphic:
            return TRUE
        #script:
            return FALSE
    end case
end isOKToAttach

on getPropertyDescriptionList me

```

```

if not the currentSpriteNum then exit

propertyDescriptionList = []
cursorTypes = []
cursorMembersList = GetCursorMembers (me)
cursorBitmapsList = GetCursorBitmaps (me)
cursorMasksList = duplicate (cursorBitmapsList)
cursorMasksList.addAt (1, "no mask")

cursorMembers = cursorMembersList.count()
bitmapCursors = cursorBitmapsList.count()
if cursorMembers then
    cursorTypes.append ("Cursor Member")
end if
if bitmapCursors then
    cursorTypes.append ("1 bit bitmap")
end if
if cursorTypes.count() then
    cursorTypes.addAt (1, "Built-in cursor")
    propertyDescriptionList.addProp \
( \
    #myCursorType, \
    [ \
        #comment: "CHOICE OF TYPE - Use which type of cursor?", \
        #format: #string, \
        #range: cursorTypes, \
        #default: cursorTypes[1]\
    ] \
)

    propertyDescriptionList.addProp \
( \
    #myBuiltInCursor, \
    [ \
        #comment: "CHOICE OF CURSOR - Built-in
cursor:", \
        #format: #cursor, \
        #default: 280\
    ] \
)

    else
        return \
[ \
    #myBuiltInCursor: \
    [ \
        #comment: "Use which cursor?", \
        #format: #cursor, \
        #default: 280\
    ] \
]
end if

```



```

    if cursorMembers then
        propertyDescriptionList.addProp \
( \
    #myCursorMember, \
    [ \
        #comment: "Cursor member:", \
        #format: #member, \
        #range: cursorMembersList, \
        #default: cursorMembersList[1] \
    ] \
)
    end if

    if bitmapCursors then
        propertyDescriptionList.addProp \
( \
    #myCustomCursor, \
    [ \
        #comment: "1 bit bitmap (image)", \
        #format: #bitmap, \
        #range: cursorBitmapsList, \
        #default: cursorBitmapsList[1] \
    ] \
)

        propertyDescriptionList.addProp \
( \
    #myCustomMask, \
    [ \
        #comment: "1 bit bitmap (mask)", \
        #format: #bitmap, \
        #range: cursorMasksList, \
        #default: cursorMasksList[1] \
    ] \
)
    end if

    return propertyDescriptionList
end

on GetCursorMembers me
    cursorMembersList = []
    maxCastLib = the number of castLibs
    repeat with theCastLib = 1 to maxCastLib
        maxMember = the number of members of castLib theCastLib
        repeat with memberNumber = 1 to maxMember
            theMember = member(memberNumber, theCastLib)
            if theMember.type = #cursor then
                if theMember.name = EMPTY then
                    cursorMembersList.append(theMember)
                else

```

```

        cursorMembersList.append(theMember.name)
    end if
end if
end repeat
end repeat
return cursorMembersList
end GetCursorMembers

```

```

on GetCursorBitmaps me
    cursorBitmapsList = []
    maxCastLib = the number of castLibs
    repeat with theCastLib = 1 to maxCastLib
        maxMember = the number of members of castLib theCastLib
        repeat with memberNumber = 1 to maxMember
            theMember = member(memberNumber, theCastLib)
            if theMember.type = #bitmap then
                if theMember.depth > 1 then next repeat
                if theMember.width > 20 then next repeat
                if theMember.height > 20 then next repeat

                if theMember.name = EMPTY then
                    cursorBitmapsList.append(theMember)
                else
                    cursorBitmapsList.append(theMember.name)
                end if
            end if
        end repeat
    end repeat
    return cursorBitmapsList
end GetCursorMembers

```

### 5.4.3 Structure Programming

Structure programming is a discipline approach to programming that results in programs that are easy to read and understand and less likely to contain errors. The emphasis is on the following accepted program style guidelines to write code that is clear and readable. Obscure tricks and structure programming shortcuts are not taught in this system package since that it is strongly discouraged. The main advantage of structure programming is that it is easier to design in the beginning and easier to maintain over the long term.

An easy to read codes makes the system package easier to be read and understand. The elements of style include (source code level) documentation, methods for data declaration



and approach to statement construction. The following are some of the used coding methods:

- Selection of meaningful identifier (variable and labels) names.
- Appropriate comments written in the source code.
- Indentation of codes increases the readability.

#### 5.4 System Documentation

The documentation of the system was started in the First phase of this software development .It was done to get a documentation that is precise, completed, quality and informative. Documentation is a kind of reference to the user who wants to learn how the system was developed .A good documentation can give a precise picture on what has occurred and taken place.

## Summary

Development strategy is very important for me to identify hardware and software that will be used in developing my training package. It allows me to complete my work without facing any major difficulties. This includes prototyping to allow user to see the real mock-up model and evaluate. Evaluation will be considered in continuing the development of the real system.

The development of the system itself by using Macromedia Director 8.0 has thought me a lot. There are many steps that have to be taken in order to get a movie that has good quality and easy to understand. Hopefully this steps that I have taken will suit the users requirements.

Coding that are used in the system package includes XML and Lingo scripts. Since XML is still in beta version, it takes a longer period to master it and present to user with confidence. While lingo scripts just take a minor role in my system – most of it was built-in script that generated by Director 8 itself.

Structure of programming allows me to present XML codes in more readable and understandable manner. It eases the ability of understanding by using variables and labels that match with its functions and meaning. Plus, the comments that were inserted throughout the entire lesson for each line of codes allow user to keep track with their training. Indentation of codes reflects the readability of programming structure.



**LIST OF REFERENCES**

- 1) Simpson, John E ,Just XML 2<sup>nd</sup> Edition,Prentice Hall
- 2) Learning XML , (Guide to) Creating Self-Describing Data ,O Reilly Publications
- 3) XML Step by Step, Prentice Hall ,2000
- 4) <http://www.gettown.edu/crossroads/mltmedia.html>
- 5) <http://tutorials.beginners.co.uk/view/cobrand/searchmiddleware/i/6>
- 6) <http://www.med.monash.edu.au/informatics/techme/authorsoft.htm>
- 7) <http://www.macromedia.com>
- 8) <http://www.adobe.com>

## CHAPTER 6 : SYSTEM TESTING

### 6.1 Introduction

In this section, the main covered would be regarding testing. It starts off by a short and simple explanation about what is testing. It then describes the testing strategy used in this project. Next, it moves on to which type of testing would be applied in this project and why they are used. It also includes the various test plans which will be drawn up when testing is implemented.

### 6.2 Testing Objectives

# CHAPTER 6

# SYSTEM TESTING

#### 6.2.1 What is testing?

Testing is the process of checking and validating the various features of the system. It is a critical element of software quality assurance and provides the ultimate review of specifications, design and coding. Testing involves Validation and Verification.

##### 1. Validation

It is used to test whether the system meets all the requirements as specified.



## CHAPTER 6 : SYSTEM TESTING

### 6.1 Introduction

In this section , the issues covered would be regarding testing. It starts of by a short and simple explanation about what is testing. It then describes the testing strategy/plan used in this project. Next, it moves on to which type of testing would be applied in this project and why they are used. It also includes the various test plans, which will be drawn up when testing is implemented.

### 6.2 Testing Objectives

The objectives of Testing are as follows:

- To identify, isolate and correct as many bugs as possible. Most programs have bugs, the most insidious of which appear only with unique combinations of data or elements.
- To demonstrate that functionality of the system appears to be working properly and performance requirements appear to be met.

#### 6.2.1 What is testing ?

Testing is the process of verifying and validating the various features of the system. It is a critical element of software quality assurance and presents the ultimate review of specifications, design and coding. Testing involves Validation and Verification.

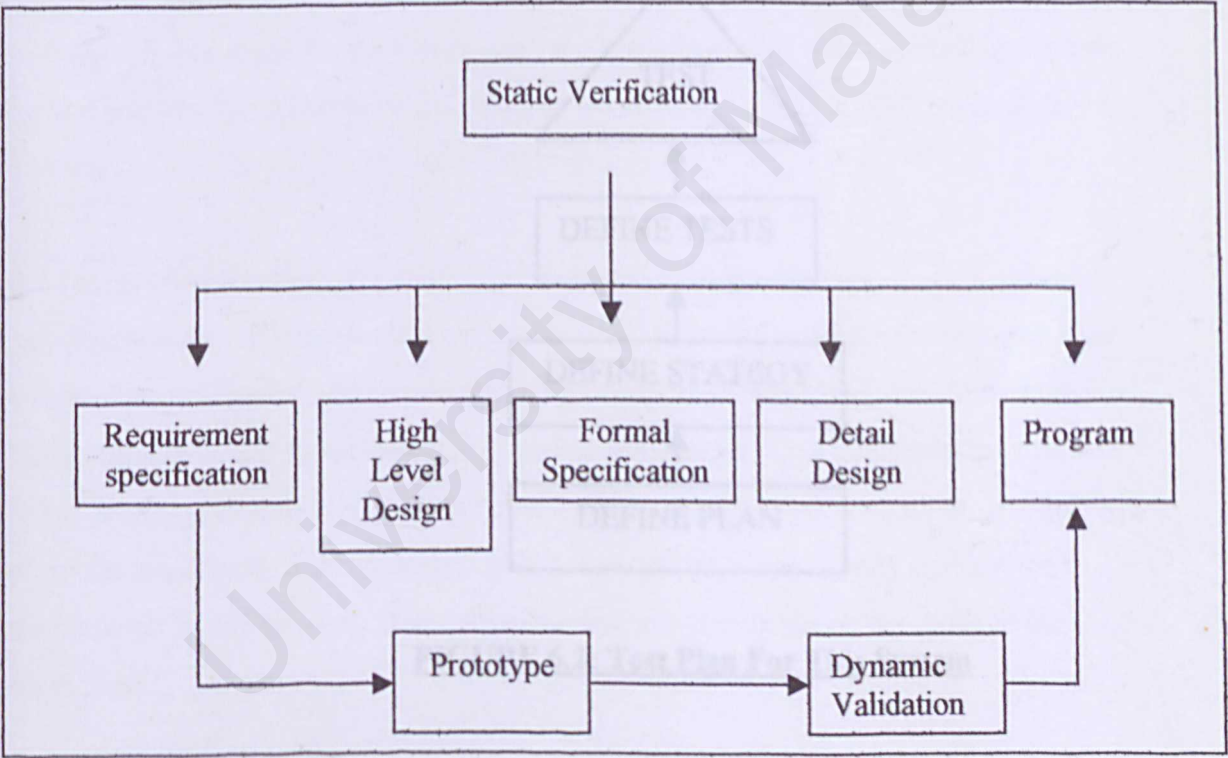
##### 1. Validation

It is used to test whether the system meets all the requirements as specified.

2. Verification

It is used to test whether the system developed meets the users requirements.

In order to meet the objective of Validation and Verification as mentioned above, techniques like Dynamic and Static Testing should be used. The Dynamic technique involves implementation exercise (Ian Sommerville, 1995) .Where else the Static technique involves analysis and testing of system performance like for example system design and system coding. This is applied in every stage through a systematic structure as shown below:

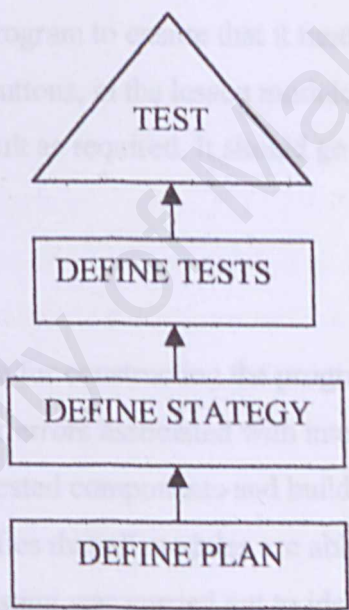


**Figure 6.1: Verification and Validation of Static and Dynamic.**



6.3 Types of testing identified in this project

Nowadays, computer software and application has become more complex. Thus, white-box and black-box testing which are applicable across the environments, architectures, and application, unique guidelines and approaches to testing are warranted. This situation/conditions are applied in the project. As such, testing for this project will be divided into several parts. The test plan strategy applied in this project is illustrated as below:



**FIGURE 6.2: Test Plan For This System**

## 6.4 Types of Testing Techniques used

In this section, the various testing techniques that will be used will be described briefly. The reason why these techniques are used are emphasized and highlighted. These testing techniques are applied in the client/server approaches when it is appropriate. The testing will begin with “testing in small” and workout towards “testing in large”. It will start with Unit Testing, followed by Integration Testing and culminated with Validation and System Testing.

### 1) Functional Testing

#### 6.4.1 Unit Testing

It concentrates on the smallest units of the software that are the modules. A module is a collection of components that are independent. Each module is tested separately.

Unit testing was tested by running the program to ensure that it functioned the way it was designed. For example, the ‘Next’ buttons, in the lesson module is tested by clicking on it and see whether it produces the result as required. It should go to the next screen in the system.

#### 6.4.2 Integration Testing

It concentrates on a systematic technique for constructing the program structure while at the same time conducting test to uncover errors associated with interfaces. The objective of integration testing is to take the unit tested components and builds a program structure that has been detected by design. It verifies that all modules are able to function together. During the integration of all modules, testing was carried out to identify the fault and failure caused by the integration as well as review and rectify the correct path of the system flow.

#### 6.4.3 Validation Testing

For validation testing, the activities that will be performed are the setting/identifying the validation test criteria, configuration review, alpha and beta testing.



#### 6.4.4 System Testing

It is a very important testing method that checks the functions of the system are proper and the overall system performance and objectives are reached. The system also ensures that it can be setup properly. The main objective is to test on the integrated system and verify it to meet the specified requirements. The behavior of the test will define the quality of the system. The system is tested on its functions, performance and the acceptance. There are several steps in testing a system. They are as below:

##### 1) Functional Testing

Functional test checks that the integrated system performs its functions as specified in the requirements.

##### 2) Performance Testing

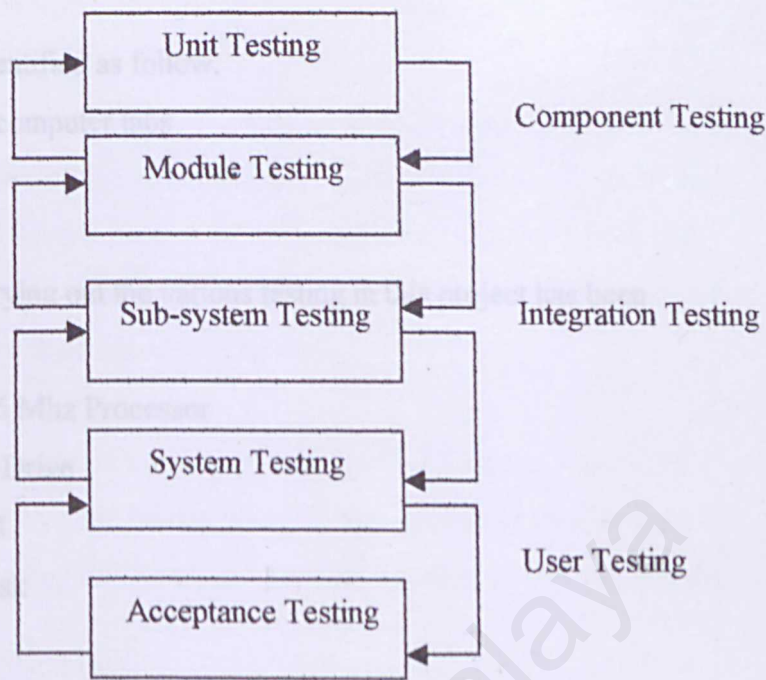
Performance testing is where function performance of the system is evaluated. It is designed to test the minimum time per frame of software within the context of an integrated system. However performance testing occurs throughout all steps at the testing process. It is not until all system elements are fully integrated that the true performance of the system can be executed.

##### 3) Acceptance Testing

The purpose of acceptance testing is to make sure the system is ready for operation. It may reveal requirement problems where the system's facilities do not meet the users' need or the system promised to users. Several copies of the system had been delivered to a number of potential users who happen to test the system and give feedback on it.

##### 4) Installation Testing

Installation testing is done to verify the system can be installed and works as tested previously especially in the environment where there are only one computer using it or even several computers attached to it.



**Figure 6.3 Components of System Testing**

**6.5 Testing Personnel**

The testing personnel identified in this project besides myself are as follow:

**6.5.1 University Malaya Students**

Testing Personnel 1

Name: Diana (Faculty of Engineering)

Testing Personnel 2

Name: Magdaleine Tang (Faculty of Science)

Testing Personnel 3

Name: Yeoh Suet Hui (Faculty of Computer Science and Information Technology)

Testing Personnel 4

Name: Harvinderjeet Singh (Faculty of Computer Science and Information Technology)



## 6.6 Testing Environment

The testing environment are identified as follow:

- University Malaya computer labs

## 6.7 Testing Hardware

The hardware required for carrying out the various testing in this project has been identified as follow:

- Pentium III 266 Mhz Processor
- 34 X CD Rom Drive
- 128 Mhz RAM
- 20 GB Hard disk
- Microphone
- Other standard desktop PC component

## 6.8 Testing Software

The software required during testing is:

- Operating System – Microsoft Windows 98
- Director 8.0
- Adobe Photoshop 6.0
- Flash 5.0
- Sound Forge and Cool Edit 2000

## Chapter Summary

Testing is very important to be done in any tasks. And for this task, testing stage is vital before the system package is delivering to the customer. Since that there will not be any maintenance services provided, testing need to be done many times to avoid from end-user to detect any errors. However, there will be different versions of the training package from time to time – continuous enhancement will be done.

The types of Testing technique used are beginning from the Unit Testing, Integration Testing, Validation Testing and System Testing which is then broken to smaller parts of testing which are Functional Testing, Performance Testing, Acceptance Testing and the finally the Installation Testing.

The final testing stage, which is to get Testing Personnel whom are different users test the system on various PCs to get precise result in reliability of the package to be playable on different platform and different Testing Environments.



## LIST OF REFERENCES

- 1) Ian Sommerville , Software Engineering , 5th Edition , Addison Wesley , 1996
- 2) Pfleeger, Shari Lawrence ,Software Engineering Theory and Practice ,N.J:Prentice Hall International Edition, 1998
- 3) S.Pressman, Roger Software Engineering A Practitioners Approach ,5th Edition, Mc Graw Hill .
- 4)Alka Jarvis and Vern Crandall , Inroads To Software Quality “ How to Guide And Toolkit, Prentice Hall PTR.

## CHAPTER 7: SYSTEM EVALUATION AND CONCLUSION

### 7.1 Introduction

This chapter looks into one of the important aspects regarding this project report. It highlights issues such as the success of the project or academic issues. It looks into the degree of success in carrying out the project, learning experience gained, things to be done differently and the value of the learning experience and how it contributes to my professional and academic expertise.

### 7.2 Problems and Solutions

# CHAPTER 7

# SYSTEM EVALUATION AND CONCLUSION

### Statement

To ensure the quality of the software, the book is designed to explain the basics of XML in order to allow all beginners to achieve a good understanding of this language well before they can proceed to a deeper understanding of the language. Besides that, I will simply give examples of codes that help the user to identify it or solve it and apply it either as Internet based or Windows based application. This will achieve my target audience in the variety.



## CHAPTER 7 : SYSTEM EVALUATION AND CONCLUSION

### 7.1 Introduction

This chapter looks into one of the important topics regarding this project report. It highlights issues such as the success of the project in academic terms. It looks into the degree of success in carrying out the project; learning experienced gained, things to be done differently and the value or the learning process and how it contributes to my professional and academic expertise.

### 7.2 Problems and Solutions

#### Problem 1: Difficulty in defining the system scope at the beginning phase

Because XML is a one of the new markup languages introduced in the market, it has very limited information on how it actually works and all the sites and books based on my research done for this language, it basically tries to explain the basics of XML and nothing in depth. Where else when I actually found one or two books that do explain more in depth of this language, this was a problem too, because this is a beginners software and users definitely would get confused with what was actually being explained to them.

#### *Solution:*

To ease the users, this software has been developed to explain the basics of XML in order to allow all beginners to actually understand this language well before they can proceed to a deeper understanding of this language. Besides that, I will simply give example of codes that helps the user to identify it by them selves and apply it either as Internet based or Windows based application. This will achieve my target audiences to be variety.

**Problem 2: Choosing suitable development tools and Programming languages.**

Another problem encountered during the system study and analysis was choosing the suitable development tool and programming languages. Many tools have been studied and analyzed. All tools have their strengths and weakness. The features of each tool was studied and then compared which is explained with more details in chapter 2 of this documentation.

**Solution**

The main source to clarify the doubts encountered is by reading the official websites of the development tools and programming languages as well as books to make sure that the development tools and the programming languages chose is the most suitable ones for my software.

**Problem 3: Lack of knowledge in the software used.**

Considering that Macromedia Director 8.0 is a new software to me, I had to take a lot of time to actually understand how things was done in Director as well as how to create files in order for me to come up with my software. All the time spend on understanding the system could be actually used to implement my system and to make it an excellent system. Besides that not all methods introduced by Director could be used due to time constraints.

**Solution**

In order to understand Director better, I had to buy a few books and surf the Internet to actually understand it perfectly , which though took a lot of my time but was slowly understood by working with Director based on the examples given in the books and the web sites.

**Problem 4: Big file size for packaged and not packaged Director 8 files**

This happened because the size of sound file is big and Director 8 does not support most of this big file though it is a multimedia element. This problem is the main reason why the computer keeps hanging while the operation was still under progression.



**Solution:**

Since smaller files improved the performance of the whole package, the sound files used are encoded in a lower sample rate and 16-bit format. This makes the sounds appear softer as well as the tone appears longer. Furthermore, the lengths of sound files are for a few seconds only to minimize the file size.

**Problem 5: Lack of font support**

Both the Macromedia Director 8 and Flash have quite a limited type of fonts. For Flash, the fonts are not displayed in a colorful way unlike the Microsoft WordArt. However, the fonts displayed are anti-alias, which makes the fonts appear non-smoothly.

**Solution:**

Develop text in the normal format. Text that was developed in Adobe Photoshop is to imply effects on it – import is needed into either Flash or Director 8. For the anti-alias, the font used should be small so that the font appears smoothly.

**Problem 6: Lack of scripting option**

Even though Director 8 itself has a scripting language – Lingo, it is not powerful enough. It does not offer more advanced programming option like a full-blown programming language. Furthermore, the mouse event listener is not fully implemented in scripting. For example, the mouse over event or mouse out event could not be implemented without using the interaction icons and duplicating the various mouse events for a button.

**Solution:**

The easiest solution is to use many icons to stimulate the various mouse events. For example, using the hotspot twice to capture the mouse over and mouse click event. Another solution is to use button to substitute the hotspot.

**Problem 7: Difficulty in creating good user interface for the learning package**

This is because there are many aspects of good designing skills that need to be met. It has to be professional with the contents that need to be presented. The various aspects includes colors, graphics, animations, navigation and, of course ease of use.

**Solution:**

Fix concepts before gather sources for designing. Get opinions and suggestions from variety users. Prototypes interface and test it to different kind of people – get comments and improve eventually. As far as I am concern, the lifecycle of designing takes a long period to end with one good ideas.

**Problem 8: Recording voice**

At first I did my recording on normal Windows voice recorder. Then I realize there are so much distortion and noises that I am not able to fix it. There are no tools in Windows voice recorder that allow me to eliminate noises and distortions.

**Solution:**

By getting advices from friends, I able to eliminate the noise minimally, but the sound are still very much low quality. I searched another voice recording software. I end up with Cool Edit that allow me to eliminate distortions and noises plus obtaining a very good sound quality that even allows me to save it as MP3 format for compressions purposes.

**Problem 9: Inserting Flash**

The first time I inserted Flash movie in Director 8, it does not functioning as well as in Flash player. The graphics, sounds and music coordination do not synchronized as how it supposed to be.

**Solution:**

I had to check each frame and scores of Director 8 and adjust its tempo for the scores to allow synchronization of music, graphics and sound coordination.



**Problem 10: The time allocated was not enough**

The time allocated for the project was not sufficient enough and this did effect the development time for my software. This was also caused by a heavy timetable and exams.

**Solution:**

A proper and systematic timetable was created in order to help me divide my time sufficiently

**Problem 11: The capacity of diskettes.**

The normal diskette available does not have the capacity to support all files used in this software.

**Solution:**

I had to actually zip all files in order to save my work and saving it into multiple numbers of disks did that.

**Problem 12: Hardware problem**

There was a big problem faced during the implementation of this software because my computer could not support Director 8.0 due to the limitation of this software. Director 8.0 uses allot of outside resources and that is the main reason why extra space is needed to allow this operation.

**Solution:**

I had to loan a computer and pay RM 500 for the duration that I used it.

### 7.3 Features and Strength

The following illustrates the key strengths of Multimedia Teaching Package:

#### i. User-friendly

Multimedia Teaching Package has a user-friendly and consistent interface for the ease of users. A set of standard GUI has been implemented and has been implemented. It also has been set to allow the users to browse in a very short time.

#### ii. Fast Response

The Multimedia Teaching Package is in CD form. All the response are depending on the speed of users' CD-ROM. Nowadays CD-ROM technology can easily run the system I developed without facing any difficulties in responding towards users navigations.

#### iii. Flexibility

The Multimedia Teaching Package provides a high degree of flexibility towards target users. The structure of the content was shaped to make sure that anyone can access the system at their own commence. Users can choose what ever module they would like based on their necessity and what ever order they would like it .

#### iv. Interactive Elements

This course managed to integrate various types of media elements. Animation and sounds also enhanced the learning of the users. Besides that it also provides a better layout and the interest of users to actually learning XML.

#### v. Consistency

Multimedia Teaching Package navigation maintain its consistencies. That includes the consistency of page design plus the flash movies as introduction of the system. These allow users to easily navigate and maintain their mood of learning through my system.



**vi. Package Contents**

Contents of my Teaching Package are technically complete and interactive that meets my main purpose – users able to understand the XML from its root to its leaf node.

**vii. User friendly**

The main important feature I tried to work towards is to make sure the system is user friendly. This is important as it allows users to navigate freely through my CD and obviously be satisfied with the way the system is created.

**7.4 System Limitations****i. Inconsistent Voice**

Voice used in the system package was recorded manually. As a result, there are flaws on my voice consistency. The depth and speed of voices are different. However, I managed to maintain my slang of language.

**ii. User without soundcard could not use my training package**

Since I explained everything in voice form, it is impossible for the user who does not have soundcard and speaker to enjoy my training of the multimedia package.

**iii. Administration Management**

Due to its design characteristics, text presented and questions in each module and lesson is generally static. This means that the contents are not changeable in the form of an executable file. This limits the scalability of dynamic scope specification in contents and questions.

## 7.5 Future Enhancement

The package could be further enhanced in the following way:

A) For the system package itself:

- The animation could be further improved with the insertion of various ActiveX control.
- Since the package is divided into subtopics, those subtopics files could be easily integrated with the Internet but the size of files should be reduced before the integration could be made.
- Another approach that I would change if given a chance is to ask more questions and seek help from more sources. I have learnt that its better to be safe and ask question then to be sorry with mistakes are hard to resolve. Especially with regards to technical issues related to programming, since help takes longer time but does not necessarily guarantee a solution.
- Given a chance I would also have started learning other multimedia tools to enhance my system better as soon as possible with hope of mastering the basics before the implementation takes place. This would be extremely beneficial.
- I would also like to wide up the scope of my software or add on more features like a search engine and so on to provide an excellent learning experience for users

B) Marketing Purposes

- Convert it into shockwave html extension to allow online marketing where the user able to access the package from the Internet.



## 7.6 Gained Knowledge and Experience

Knowledge and experience has been gained throughout the development of this project. Among the are:

- Experience in learning a new programming languages outside the formal classroom.
- Enhanced creativity and initiative in developing the most appropriate system for end-user.
- Sharpened problem-solving skill by finding solution to all problems faced during the development of the project.
- Practicing proper time management to divide the time appropriately among all the subjects taken, and not neglecting any of them.
- Improved confidence in own ability.
- Willingness to listen and evaluate other opinions and correct oneself.
- These experiences will be used as “weapons” to stand out in the competitive field of multimedia that would only lead me to a bright, successful career in the near future.
- An extremely important hand-on experience has been gained throughout the creating process of this software. Software development would be easy if there are no changes as in walking on water if it is frozen. The statement really hit home with me.

## Summary

The development of this system is not just to fulfill the requirements for a degree, but also as an experience for future use. This will be helpful when I have to face with the task of developing another multimedia system. The exercise is hoped to inspire confidence in the student capability.

Even though this multimedia training package of XML was developed without applying much complicated coding, I still need to present the latest language that not all people know about it, and it takes longer period for me to master XML besides preparing on how to present it. Moreover, it needs high sense of creativity to fulfill all the requirements of a multimedia system. Most of my designing timeframe was spent on choosing appropriate graphics, recording voice in order to obtain an attractive system interface to the end-user. Their needs and desires were taken into considerations in making sure that this system would be beneficial towards them.

The overall conclusion is that the project has been carried out successfully. However the results would have been better if given more time to do and apply the knowledge and experience gained from this one. Hopefully the effort and time spent on this project is fruitful. Last but not least, I am thankful and grateful for the valuable learning experience this project has given me. These experiences have influenced me to look at things from various perspectives and more knowledge had been gained both academically and professionally.



## USER MANUAL

### 1.0 INTRODUCTION

This user manual is to help users to use and to run the multimedia teaching package without any problems. The multimedia teaching package 'Learning XML' was developed to help users to learn the Extensible Markup Language (XML) at their own pace and comfort. This teaching package can be used as Learning Aid in colleges or schools to enhance the usual blackboard technique. This multimedia Teaching Package has multimedia elements such as pictures, text, audio, video and animation to make the learning process more interesting.

# USER MANUAL

### 1.1 SYSTEM CHARACTERISTICS

- The information about XML is very simple. It is for beginners who want to learn the language.
- This multimedia teaching package is easy to navigate with the buttons and icons provided.
- It uses English as the presentation language which is universal.
- It has examples of the codes that are used in XML which were thought in the teaching package.

## **USER MANUAL**

### **1.0 INTRODUCTION**

This user manual is to help users to use and to run the multimedia teaching package without any problems. The multimedia teaching package 'Learning XML' was developed to help users to learn the Extensible Markup Language (XML) at their own pace and comfort. This teaching package can be used as Learning Aid in colleges or schools to remove the usual blackboard technique. This multimedia Teaching Package has multimedia elements such as graphics, text, audio, video and animation to make the learning process more interesting.

### **2.0 USER MANUAL EXPLANATION**

- This manual is presented in a simple and easy format in English
- This manual explains the characteristics of the multimedia teaching package
- The explanation about hardware and software requirement is given
- The manual contains the modules segmentation
- The explanation about the interfaces of the teaching package

### **3.0 SYSTEM CHARACTERISTICS**

- The information and lessons about XML is very simple. It is for beginners who want to learn the language.
- This multimedia teaching package is easy to navigate with the buttons and icons provided.
- It uses English as the presentation language which is universal
- It has example of the codes that are used in XML which were thought in the teaching package



## 4.0 USAGE STEPS

Before you run the multimedia teaching package you must make sure that you have the minimum requirements :

### 4.1 Hardware Requirements

- Any Pc that has Pentium II or a higher speed computer
- At least 30 MB of free space in your hard disc
- CD-Rom Driver
- 64MB Ram
- Speakers
- Mouse

### 4.2 Software Requirements

Since the package is in .exe format it can run on any pc with windows

### Steps to run Learning XML-CD

- 1) Insert the CD into the CDROM driver
- 2) From Windows Explorer ,click on the directory D:XML(CDROM)
- 3) Click twice on the icon with the XML\_Package to run the program

## 5.0 Modules

Flash Introduction

Main Menu

There are 5 modules on the main menu interface

- 1) Introduction
- 2) Lessons
- 3) Tutorials
- 4) Library
- 5) Appendix

## 6.0 THE INTERFACES IN THE MULTIMEDIA TEACHING PACKAGE

- 1) After clicking twice on the icon XML\_package, the flash introduction will appear.  
Right after that the main page for Learning XML package comes out.



From here users can make a choice to choose on what module that they would like to start on.



2) If they click on Introduction a Flash Introduction will appear. As soon as that is over the Introduction screen will appear.

## INTRODUCTION LEARNING XML

XML emerged as a way to overcome the shortcomings of its two predecessors, SGML and HTML which were both very successful markup languages, but which were both flawed in certain ways. SGML the international standard for marking up data, has been used since the 80s. SGML is an extremely powerful and extensible tool for semantic markup which is particularly useful for cataloging and indexing data. However, SGML is complex, especially for the everyday uses of the web. Not only that, but SGML is pretty expensive. Adding SGML capability to a word processor could double or triple the price. Finally, the commercial browsers made it pretty clear that they did not intend to ever support SGML. HTML on the other hand was free, simple and widely supported. HTML was originally designed at CERN around 1990 to provide a very simple version of SGML which could be used by "regular" people. As everyone knows, HTML spread like wildfire. Unfortunately, HTML had serious defects.

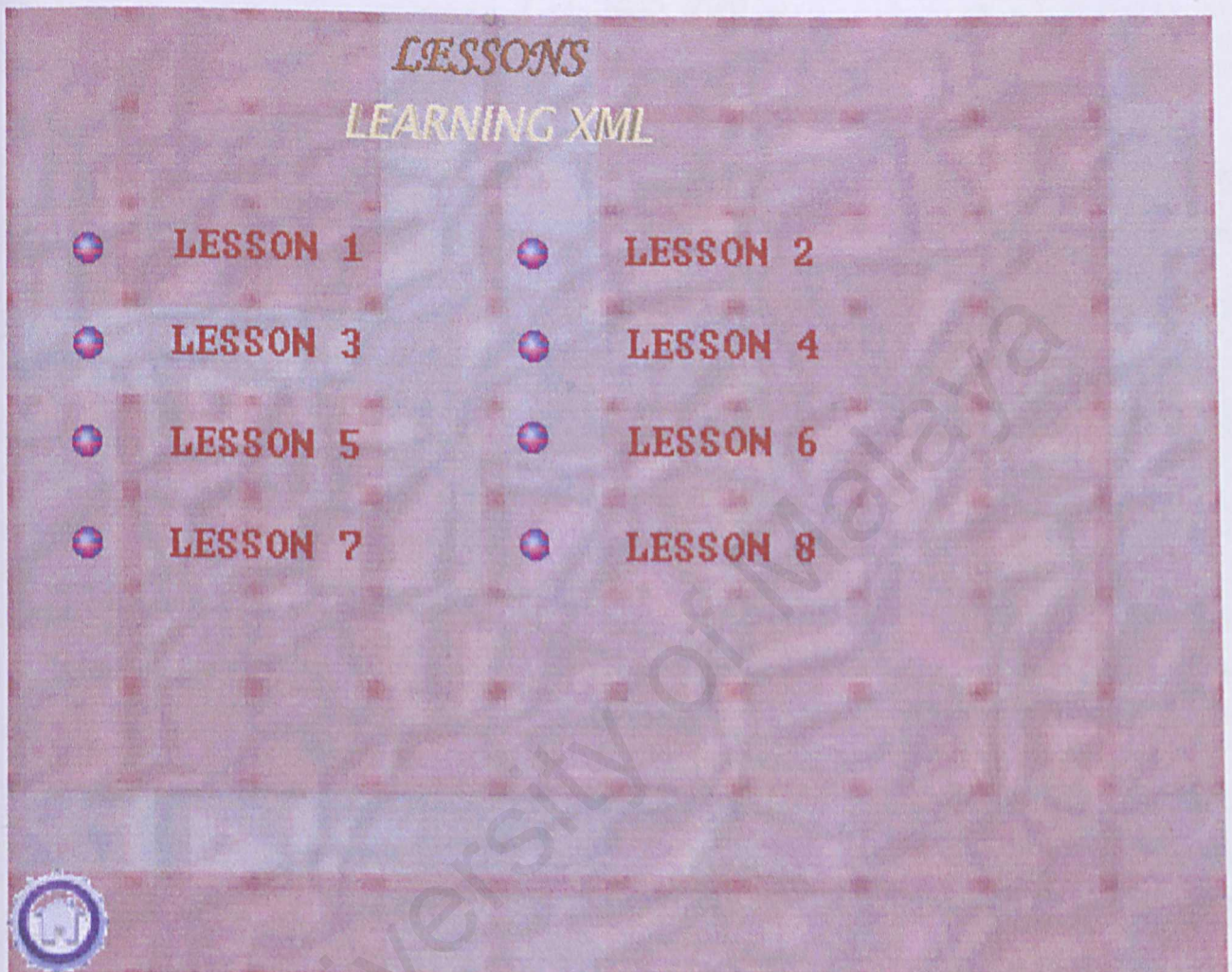
So in 1996, discussions focused on how to define a markup language with the power and extensibility of SGML but with the simplicity of HTML. The W3C sponsored a group of SGML gurus including Jon Bosak from Sun.

All of the non-essential, unused, cryptic parts of SGML were sliced away. What remained was a lean, mean marking up machine: XML. The specification of XML (written mostly by Tim Bray and C.M. Sperberg-McQueen) was only 26 pages as opposed to the 500+ pages of the SGML specification. Nevertheless, all the useful things which could be done by SGML, could also be done with XML. Over the next few years, XML evolved, drawing from the work of its sponsors and the work of developers solving similar problems. By the summer of 1997, Microsoft had launched the Channel Definition Format (CDF) as one of the first real-world applications of XML. Finally, in 1998, the W3C approved Version 1.0 of the XML specification and a new language was born.



If they click on the home button it will go to the main page and if they click on the Next arrow it will go on the next screen

If users click on the Lessons icon on the main page ,the this page will appear.



From here users can choose on what lesson that they want.



If they Click on Lesson 1, this page will appear.

LEARNING XML

LESSON 1

# WELL-FORMED XML

Well formed XML is xml meets certain grammatical rules outlined in the XML 1.0 specification.

Tags and Text and Elements

```
<name>
  <first>John</first>
  <middle>Fitzgerald Johansen</middle>
  <last>Doe</last>
</name>
```

The words between the < and > characters are XML tags. The information in our document (our data) is contained within the various tags that constitute the markup of the document. This makes it easy to distinguish the information in the document from the markup. As you can see, the tags are paired together, so that any opening tag also has a closing tag. In XML parlance, these are called start-tags and end-tags. The end-tags are the same as the start-tags, except that they have a "/" right after the opening < character.

All of the information from the start of a start-tag to the end of an end-tag, and including everything in between, is called an element. So:

<first>

 is a start-tag

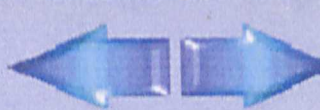

</first>

 is an end-tag

<first>John</first>

 is an element

The text between the start-tag and end-tag of an element is called the element content. The content between our tags will often just be data (as opposed to other elements). In this case, the element content is referred to as Parsed Character DATA, which is almost always referred to using its acronym, PCDATA. Whenever you come across a strange-looking term like PCDATA, it's usually a good bet the term is inherited from SGML. Because XML is a subset of SGML, there are a lot of these inherited terms. The whole document, starting at <name> and ending at </name>, is also an element, which happens to include other elements.



If they click on the home button it will go to the main screen. If they click on the next arrow it will go on to the next screen. This goes for all the 8 Lessons.



If the users Click on the Tutorial icon on the main page , This page will appear:



# TUTORIALS

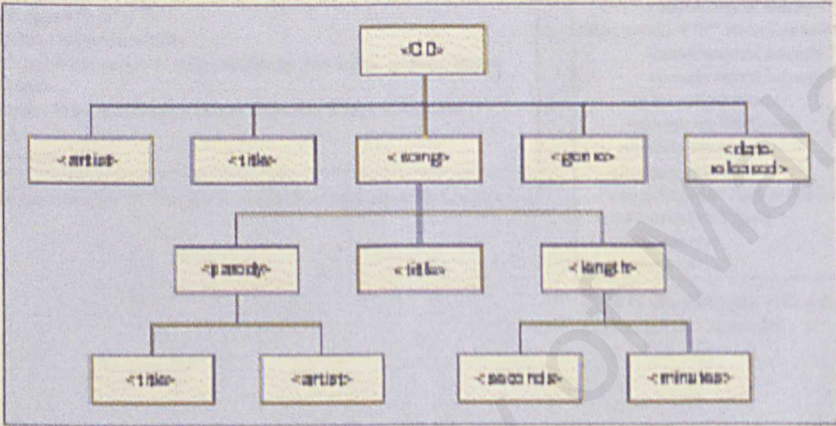
## EXAMPLE 1

We're going to build an XML document to describe a CD that was produced, Dare to be Stupid, by Weird Al Yankovic. But before we break out Notepad and start typing, we need to know what information we're capturing.

XML is hierarchical in nature; information is structured like a tree, with parent/child relationships. This means that we'll have to arrange our CD information in a tree structure as well.

Since this is a CD, we'll need to capture information like the artist, title, and date released, as well as the genre of music. We'll also need information about each song on the CD, such as the title and length. And, since Weird Al is famous for his parodies, we'll include information about what song (if any) this one is a parody of.

Here's the hierarchy we'll be creating:



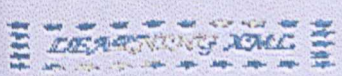
Some of these elements, like <artist>, will appear only once; others, like <song>, will appear multiple times in the document. Also, some will have PCDATA only, while some will include their information as child elements instead. For example, the <artist> element will contain PCDATA for the title, whereas the <song> element won't contain any PCDATA of its own, but will contain child elements that further break down the information.



If users Click on the Home button it goes to the main page, where else the next arrow will go on to the next screen. The previous arrow goes to the previous screen.



If users click on the Library icon on the main page ,users will see this screen:



# LIBRARY

```
<?xml version='1.0' encoding='utf-8'?>
<!-- A SAMPLE set of slides -->
<slideshow
  title="Sample Slide Show"
  date="Date of publication"
  author="Yours Truly"
>
  <!-- TITLE SLIDE -->
  <slide type="all">
    <title>Wake up to WonderWidgets!</title>
  </slide>
  <!-- OVERVIEW -->
  <slide type="all">
    <title>Overview</title>
    <item>Why <em>WonderWidgets</em> are great</item>
    <item/>
    <item>Who <em>buys</em> WonderWidgets</item>
  </slide>
</slideshow>
```

This is an example of the XML codes that was done in Lesson 4

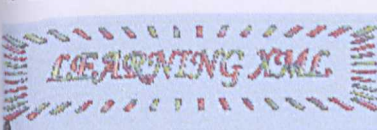
```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="book">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="title" type="xs:string"/>
        <xs:element name="author" type="xs:string"/>
        <xs:element name="character" minOccurs="0"
maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:sequence>
    <xs:element name="name" type="xs:string"/>
    <xs:element name="friend-of" type="xs:string"
minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="since" type="xs:date"/>
    <xs:element name="qualification" type="xs:string"/>
  </xs:sequence>
  <xs:complexType>
    <xs:sequence>
      <xs:attribute name="isbn" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

This is the example codes that are using Schema which was discussed in Chapter 6



If users Click on the Home button it goes to the main page, where else the next arrow will go on to the next screen. The previous arrow goes to the previous screen.

If users click on the Appendix icon on the main page ,users will see this screen :



# APPENDIX

**archiving**  
Saving the state of an object and restoring it.

**attribute**  
A qualifier on an XML [tag](#) that provides additional information. For example, in the tag <slide title="My Slide">, title is an attribute, and My Slide is its value.

**B**

**binary entity**  
See [unparsed entity](#).



**binding**  
Construction of the code needed to process a well-defined bit of XML data.

**C**

**comment**  
Text in an XML document that is ignored, unless the parser is specifically told to recognize it. A comment is enclosed in a comment tag, like this: <!-- This is a comment -->

**content**  
The part of an XML document that occurs after the [prolog](#), including the [root](#) element and everything it contains.

**CDATA**  
A predefined XML tag for "Character DATA" that says "don't interpret these characters", as opposed to "Parsed Character Data" (PCDATA), in which the normal rules of XML syntax apply (for example, angle brackets demarcate XML tags, tags define XML elements, etc.). CDATA sections are typically used to show examples of XML syntax. Like this:  
<![CDATA[ <slide>.A sample slide..</slide> ]P>  
which displays as:  
  
<slide>.A sample slide..</slide>



If users Click on the Home button it goes to the main page, where else the next arrow will go on to the next screen.