

8-16-1996

Design and Implementation of a Model for Authoring and Presenting Interactive Multimedia Documents

Vibha Mukul Nayyar
Portland State University

Follow this and additional works at: https://pdxscholar.library.pdx.edu/open_access_etds



Part of the [Electrical and Computer Engineering Commons](#)

Let us know how access to this document benefits you.

Recommended Citation

Nayyar, Vibha Mukul, "Design and Implementation of a Model for Authoring and Presenting Interactive Multimedia Documents" (1996). *Dissertations and Theses*. Paper 5187.

<https://doi.org/10.15760/etd.7063>

This Thesis is brought to you for free and open access. It has been accepted for inclusion in Dissertations and Theses by an authorized administrator of PDXScholar. Please contact us if we can make this document more accessible: pdxscholar@pdx.edu.

THESIS APPROVAL

The abstract and thesis of Vibha Mukul Nayyar for the Master of Science in Electrical and Computer Engineering were presented August 16, 1996, and accepted by the thesis committee and the department.

COMMITTEE APPROVALS:

[Redacted Signature]

Michael A. Driscoll, Chair

[Redacted Signature]

W. Robert Daasch

[Redacted Signature]

Andrew P. Tolmach
Representative of the Office of Graduate Studies

DEPARTMENT APPROVAL :

[Redacted Signature]

Rolf Schaumann, Chair
Department of Electrical Engineering

ACCEPTED FOR PORTLAND STATE UNIVERSITY BY THE LIBRARY

by

[Redacted Signature]

on 30 Sept 1996

ABSTRACT

An abstract of the thesis of Vibha Mukul Nayyar for the Master of Science in Electrical and Computer Engineering presented August 16, 1996.

Title: Design and implementation of a model for authoring and presenting interactive multimedia documents

Multimedia technology enables direct manipulation of multiple media such as text, image, audio, and video, all integrated into one entity - the multimedia document. Interactive multimedia documents integrate text, images and continuous media such as audio and video, treat them as objects to be presented for a certain duration, and allow users to interact with the presentation. Authoring and presenting interactive multimedia documents imposes new requirements on document representation. We designed the interactive timeline model (ITM) for authoring and presenting interactive multimedia documents. We implemented the model and created a playback tool using the scripting language Tcl.

ITM uses an enhanced timeline model for representing interactive multimedia scenarios. ITM is a layered model that supports the creation of

the logical, temporal, and presentation structures of a multimedia document and emphasizes the separation of content from its structure. The storage layer provides interfaces for creation and storage of media objects like text, images, scripts, choices, and audio. Choice is a media object that allows users to interact with the presentation. A script object causes Tcl code to be executed. The composition layer specifies the logical structure and provides interfaces to compose a multimedia document. The presentation layer specifies the temporal and spatial structure of the document and describes the user interface for the playback tool. The run-time layer specifies the interfaces to start and control the flow of the presentation, describes the possible interactions between a user and the presentation at run-time, and specifies the actions to be taken when interaction takes place.

ITM allows users to configure the playback tool according to their needs. It provides authors with a mechanism to create new interfaces and register them with the playback tool. To evaluate the design decisions of ITM and verify its usability, we conducted usability tests on the model. We asked a representative sample of end users (both authors and viewers) to perform realistic tasks using the model. ITM is a powerful model to create and present interactive learning materials and dynamic presentations.

DESIGN AND IMPLEMENTATION OF A MODEL FOR AUTHORIZING
AND PRESENTING INTERACTIVE MULTIMEDIA DOCUMENTS

by

Vibha Mukul Nayyar

A thesis submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE
in
ELECTRICAL AND COMPUTER ENGINEERING

Portland State University
1996

TABLE OF CONTENTS

TABLE OF FIGURES.....	vi
------------------------------	-----------

TABLE OF TABLES.....	vii
-----------------------------	------------

CHAPTERS

1. INTRODUCTION.....	1
1.1 MOTIVATION AND OBJECTIVE.....	1
<i>1.1.1 The need for a multimedia document model.....</i>	<i>1</i>
<i>1.1.2 Multimedia document models.....</i>	<i>2</i>
<i>1.1.3 Document model for interactive multimedia.....</i>	<i>4</i>
1.2 THE INTERACTIVE TIMELINE MODEL.....	5
<i>1.2.1 An Overview.....</i>	<i>5</i>
<i>1.2.2 The ITM model.....</i>	<i>6</i>
<i>1.2.3 The layers of ITM model.....</i>	<i>10</i>
<i>1.2.4 ITM Evaluation.....</i>	<i>11</i>
1.3 USES OF ITM.....	12
<i>1.3.1 ITM as an authoring tool.....</i>	<i>12</i>
<i>1.3.2 ITM for distance education.....</i>	<i>12</i>
2. MULTIMEDIA DOCUMENT MODELS.....	19

2.1 AN OVERVIEW	19
2.2 DOCUMENT MODELS.....	20
2.2.1 <i>What is a document model ?</i>	20
2.2.2 <i>Structuring traditional documents</i>	21
2.2.3 <i>Structuring multimedia documents</i>	22
2.3 MODELING MULTIMEDIA DOCUMENTS	25
2.3.1 <i>Terminology</i>	25
2.3.1.1 Multimedia Document Model	25
2.3.1.2 Multimedia Scenario	25
2.3.1.3 Logical structure	26
2.3.1.4 Temporal Structure	26
2.3.1.5 Presentation structure	27
2.3.1.6 Control structure.....	29
2.3.2 <i>Different views of a multimedia document</i>	29
2.4 MODELING TIME IN A MULTIMEDIA DOCUMENT.....	33
2.4.1 <i>Events</i>	33
2.4.2 <i>Temporal Specification</i>	33
2.4.3 <i>Synchronous and Asynchronous Events</i>	34
2.4.4 <i>Temporal Equality</i>	34

2.4.5 <i>Models of Time</i>	35
2.5 REQUIREMENTS OF THE MODEL FOR INTERACTIVE MULTIMEDIA DOCUMENTS ..	37
2.5.1 <i>Document structure requirements</i>	37
2.5.1.1 Logical structure.....	37
2.5.1.2 Temporal structure.....	43
2.5.1.3 Presentation structure	47
2.5.1.4 Control Structure	48
2.5.2 <i>The Model as a Software Product</i>	49
3. THE INTERACTIVE TIMELINE MODEL	58
3.1 AN OVERVIEW	58
3.2 DESIGN GOALS.....	58
3.3 THE INTERACTIVE TIMELINE MODEL	60
3.3.1 <i>An Overview</i>	60
3.3.2 <i>The layered ITM model</i>	61
3.3.2.1 Storage layer.....	61
3.3.2.2 Composition layer	63
3.3.2.3 Presentation layer	63
3.3.2.4 Run-time layer	64
3.3.3 <i>Document structure</i>	65
3.3.3.1 Logical Structure	65

3.3.3.2 Temporal Structure	69
3.3.3.3 Presentation Structure.....	73
3.3.3.4 Control Structure	76
3.4 INTERFACES OF ITM	78
3.5 FEATURES OF ITM	82
3.6 LIMITATIONS OF ITM	85
4. EVALUATION OF ITM	87
4.1 AN OVERVIEW	87
4.2 BACKGROUND ON USABILITY TESTING	88
4.2.1 <i>An Overview</i>	88
4.2.2 <i>Usability</i>	88
4.2.3 <i>Usability Testing</i>	89
4.3 ITM EVALUATION - OBJECTIVES AND METHODOLOGY	92
4.3.1 <i>Evaluation Objectives</i>	92
4.3.2 <i>Evaluation Methodology</i>	93
4.3.2.1 Exploratory Test	94
4.3.2.2 Comparison Test.....	95
4.3.2.3 Assessment Test	96
4.3.2.4 Validation Test	97

4.4 EVALUATION RESULTS.....	101
5. CONCLUSIONS	103
5.1 THE INTERACTIVE TIMELINE MODEL	103
5.2 ITM EVALUATION	106
5.3 FUTURE WORK.....	107
REFERENCES.....	110

TABLE OF FIGURES

FIGURE 1.1 : THE BASIC TIMELINE MODEL.....	3
FIGURE 1.2 : A TIMELINE MODEL THAT INCLUDES CHOICE OBJECTS.....	5
FIGURE 1.3 : A TIMELINE REPRESENTATION OF ITM.....	9
FIGURE 1.4 : THE LAYERED ITM MODEL SHOWING THE LAYERS OF ITM AND THE ABSTRACTIONS PROVIDED BY EACH LAYER	11
FIGURE 2.1 : EXAMPLE OF AN OFFICE DOCUMENT ARCHITECTURE STRUCTURED DOCUMENT	24
FIGURE 2.2 : A HIERARCHICAL REPRESENTATION OF DIFFERENT VIEWS OF AN INTERACTIVE MULTIMEDIA DOCUMENT	30
FIGURE 2.3 : HOW THE PETRI NET MODEL WOULD REPRESENT THE SCENARIO PRESENTED	41
FIGURE 2.4 : CMIFed CHANNEL VIEW	42
FIGURE 3.1 : THE LAYERED ITM MODEL	65
FIGURE 3.2 : LOGICAL STRUCTURE ABSTRACTIONS.....	66
FIGURE 3.3 : TIMELINE REPRESENTATION OF ITM	66
FIGURE 3.5 : TEMPORAL STRUCTURE SUPPORTED BY ITM.....	72
FIGURE 3.4 : VARIOUS LINKS SUPPORTED BY ITM.....	74
FIGURE 3.6 : THE USER INTERFACE	76
FIGURE 3.7 : EXAMPLE OF UNIT-DEFINITION FILE.....	81

TABLE OF TABLES

TABLE 1.1 INTERACTION STYLES IN MMIS	17
TABLE 2.1 : DIFFERENT VIEWS OF A MULTIMEDIA DOCUMENT	31
TABLE 3.1 : FUNCTION OF BUTTONS ON THE USER INTERFACE	75
TABLE 3.2 AUTHOR AND SUPPORT INTERFACES PROVIDED BY ITM.....	79
TABLE 3.3 FEATURES AND IMPLEMENTATION DETAILS FOR ITM.....	85

CHAPTER 1

1. INTRODUCTION

1.1 Motivation and Objective

1.1.1 The need for a multimedia document model

Multimedia information technology and applications have advanced rapidly in recent years. Multimedia application development requires a specific combination of a hardware platform and authoring software. Multimedia technology enables direct manipulation of new types of information such as image, audio, and video, all integrated into one entity - the multimedia document. Multimedia documents differ from traditional documents composed of text and geometric graphics. Multimedia documents contain multiple media, which may be presented simultaneously or in some related manner in time.

Authoring a multimedia document involves creation of multimedia scenarios, which are fully specified temporal entities involving multiple media. In view of the existing variety of multimedia authoring systems, the challenge lies in providing an integrated way to access, process, and communicate multimedia information. In this context, we require a model that allows creation, storage, and

presentation of multimedia data in a clear and concise manner, yet is powerful enough to describe multimedia documents to the extent required by the authors.

1.1.2 Multimedia document models

Current research on multimedia document representation can be divided into two areas - one focusing on passive multimedia documents, the other on active multimedia documents [1]. In passive multimedia documents, the author integrates continuous media, such as audio or video sequence in a static visual form, later activated by the user. An active multimedia document also integrates continuous media, but treats them as objects to be presented for a certain duration. It does not allow user interaction with the presentation. Interactive multimedia documents are a special case of active multimedia documents. An interactive multimedia document not only integrates and presents continuous media, but also allows the user to interact with the running presentation.

The inclusion of continuous media in active and interactive multimedia documents imposes new requirements on document representation and storage. The model for an interactive multimedia document should not only define the organization of the contents of the document (logical structure), but also specify when the contents will be presented relative to a reference point in time (temporal

structure), how the presentation of contents will be synchronized, how the contents will appear on the display device (presentation structure), and how the interaction of the user with the presentation will be controlled (control structure).

Perhaps the most prevalent model for active multimedia documents is the traditional timeline model. In this model, multiple media are presented along a single axis representing time (Figure 1.1). Though this model is simple, and graphical, it lacks the flexibility to represent relations that are determined interactively, such as at run time. Several models have been developed to represent a multimedia document [1-2, 4-5]. As explained in Chapter 2, none of these models fully satisfies the requirements of an interactive multimedia document model.

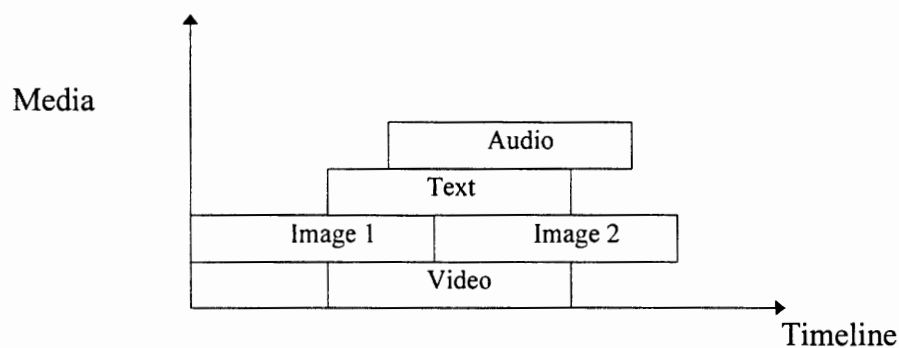


Figure 1.1 : The Basic Timeline Model

1.1.3 Document model for interactive multimedia

A model that provides a way to represent interactive multimedia scenarios that not only play-back, but also change course dynamically depending on user interactions, is described by Karmouch et al in [2]. This model extends the traditional timeline model to accommodate interactivity in a presentation. It introduces a new media type called choice, that increases the power of the timeline (Figure 1.2). "Choice" is placed directly on the timeline, and presents the user with different options to change the course of the presentation. For example, when a "choice" is presented, the user can make a selection to jump to another timeline. This model also provides a way to represent "asynchronous events". These are events whose start time is known, but whose end time, and hence, duration cannot be known in advance.

The model presented in [2] is not flexible enough to handle the indeterminism that arises from user interaction with a presentation. For example, if the user does not make a selection during a choice object's duration, he loses the chance to make it. The presentation continues from the destination timeline, and the original scenario terminates. The manner in which asynchronous events are modeled does not allow any user interaction. Once the user makes a choice, an

asynchronous event is activated, but it is deactivated only when the presentation flows to a new timeline. Also, this model does not address the spatial layout of the document and the implementation issues of the model.

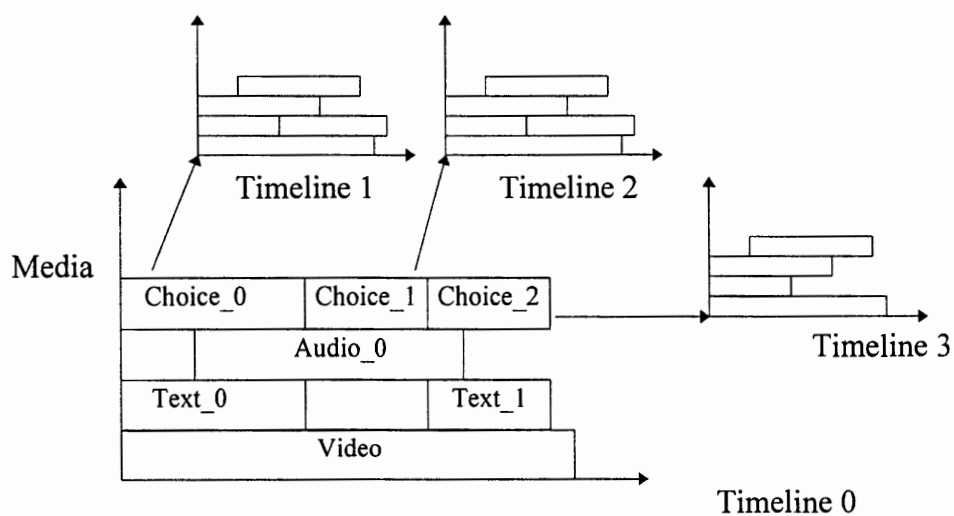


Figure 1.2 : A timeline model that includes choice objects

1.2 The Interactive Timeline model

1.2.1 An Overview

We propose to extend the model described in [2] to include features that increase the level of user interaction with the multimedia document presentation.

We present the interactive timeline model (ITM) for interactive multimedia

scenarios. A contribution of this model is that it establishes the relationship between the logical, temporal, presentation, and control structures of an interactive multimedia document. It emphasizes the separation of multimedia data from its temporal structure, and its presentation structure. In this work, we present the ITM model, its features, and uses. We implement the features of the model using the scripting language, Tcl [17], and create a playback tool for presenting interactive multimedia documents. We also evaluate the model from the authors' and viewers' perspective, by conducting tests on the model, and its prototype implementation. The following sections give an introduction to the ITM.

1.2.2 The ITM model

The interactive timeline model (ITM) uses the model described in [2] as the starting point and modifies and extends the model to increase user interaction with the multimedia presentation. A timeline representation of the ITM model is given in Figure 1.3.

The following original modifications have been done to the original model:

1. The ITM model provides the author with various choice types, such as options to inform the user to make a particular selection using a dialog box, or a text box or by clicking on a default or author-defined button etc.
2. Each choice has multiple responses at run-time. This will allow the user to seek more complex answers to questions, each time he makes the same selection.
3. We have added a new media type called "script", that allows the author to run Tcl programs during a presentation.

A media type "text object", allows authors to create text files with hypertext links. It also allows the author to embed Tcl code in the text to be presented. We call this "active text".

5. Each timeline is divided into a number of "units". Each unit has a fixed duration (length of the unit), and represents a multimedia scenario that consists of the media objects - text, images, audio, multiple scripts, and multiple choices. Units are required to support the ITM model, as the length of the unit is used to synchronize the presentation of the objects within a unit.
6. We control the interaction of the user with the presentation by providing a control structure to the document. This is the interface that takes control from the user, in case he makes an incorrect choice. For example, if the user is presented

with a choice to click on a particular region of the screen, and the user clicks elsewhere, the interface waits for the correct response for a fixed time, and thereafter, acts on behalf of the user.

7. The ITM model provides different types of links for user interaction. The user can navigate through the document by means of three types of links - text links, unit links, and timeline links. Text links allow the user to jump to a portion of text within the same unit. Unit links allow the user to jump to a different unit within the same timeline. Timeline links allow the user to jump from a timeline to any other timeline during a presentation.

8. To support asynchronous events, we modify the original model, by dividing the timeline into units with a fixed, but arbitrary duration. Thus, the model allows the user to interact with asynchronous events. It also eliminates the indeterminism related to asynchronous events, since it forces the event to end with the unit. Hence, the maximum duration of such events is bounded by the length of the unit. If a media object is located on a remote database, the author cannot create a link to it without knowing how long it would take to present it. This is a limitation of the ITM model. The author must know, in advance, exactly how long it would take to present the object. Only if the duration of the presentation is less than or equal to the length of the unit, the author can include the object into the unit. An

alternative solution is that the author can adjust the length of the unit to accommodate this object.

9. The ITM is a layered model that separates the definition, composition, and presentation of media objects in an interactive multimedia document. Figure 1.4 shows the layered ITM model. The separation of data definition, composition, and presentation layers allows for reusability of data as the same data objects can be shared by several presentation structures. Section 1.2.3 gives an introduction to each layer of the model.

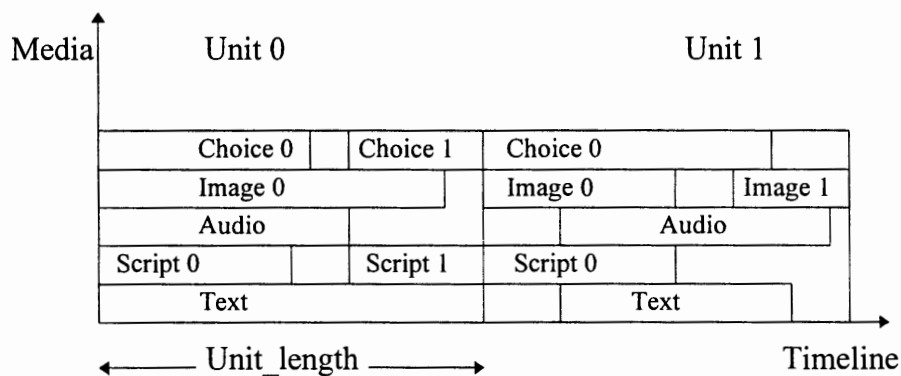


Figure 1.3 : A timeline representation of ITM

1.2.3 The layers of ITM model

1. Storage Layer - This layer allows the specification of data, and provides abstractions of data objects for the higher layers. It provides interfaces for creation of choice media objects.

2. Composition Layer - This layer specifies the logical structure of the contents of the document, and provides interfaces for grouping the data objects defined in the data definition layer into a multimedia scenario.

3. Presentation Layer - This layer provides a description of when and how the data is to be presented on the display device, and how the media objects will be synchronized during presentation. It also describes the user interface for the playback tool.

4. Run-time Layer - This layer specifies the interfaces to start and control the flow of the presentation. This layer also describes the possible interactions between a user or an application and the presentation at run-time.

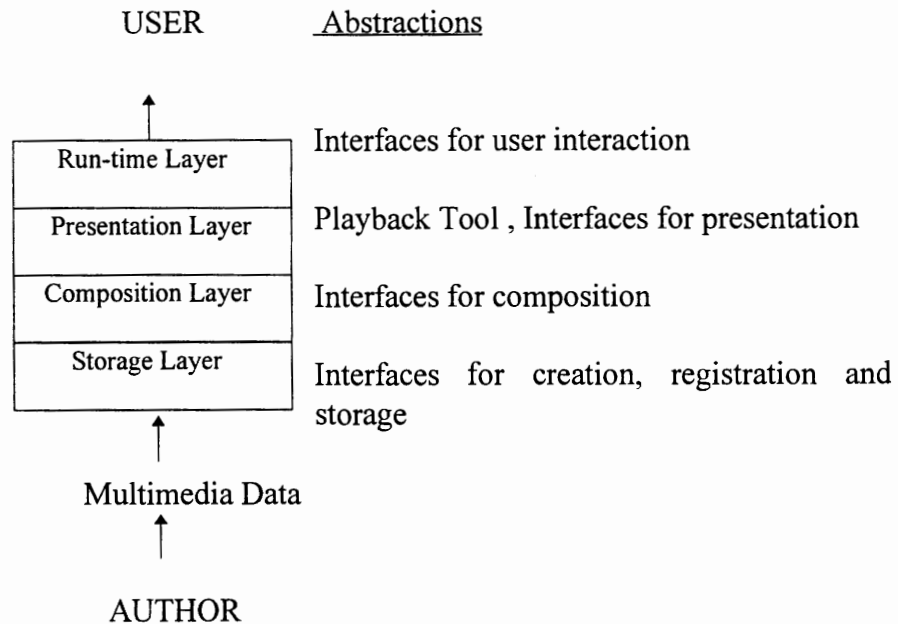


Figure 1.4 : The layered ITM model showing the layers of ITM and the abstractions provided by each layer

1.2.4 ITM Evaluation

To evaluate the design decisions of ITM, and to get an idea about the usability of the model, we evaluated ITM. We adopted an informal approach to evaluation of ITM. We conducted tests using a small sample of representative end users (2 authors and 3 viewers) to get feedback on the design of ITM and the usefulness of the playback tool. We evaluated the model by asking authors to create an interactive tutorial for an X-windows based application. Also, we got

feedback from the authors to improve the design of ITM. We asked users to view the tutorial and evaluate the playback tool.

1.3 Uses of ITM

1.3.1 ITM as an authoring tool

We believe that the interactive timeline model can be effectively used as an authoring tool to create interactive multimedia documents for various applications. Using this model to create interactive instructional course material is probably the best application of ITM. Not only can the author create multimedia tutorials with ITM, but the users get the chance to interact with the presentation and change its course. Authors can create interactive training materials, books, and encyclopedias. Another useful application is creating interactive news articles for multimedia news-on-demand.

1.3.2 ITM for distance education

Our research on the role of interactive multimedia in distance learning shows that multimedia has the potential to extend the information technology methods previously established by interactive learning modules in teaching. Multimedia, used innovatively, has strong attractions for both teachers and students of all levels across a wide variety of subjects.

Active learning multimedia environments enable students to make authentic choices that impact the learning environment [14]. Several case studies to illustrate the advantages of truly interactive multimedia technologies like multimedia simulations, tutors, systems for composition and collaboration, and explanatory systems, are presented in [15]. These authors assert that multimedia, when coupled with other improvements in educational software design, can support effective quality instruction. Collaborative learning is a process that emphasizes group or cooperative efforts among faculty and students, active participation and interaction on the part of both students and instructors, and new knowledge that emerges from an active dialog among those who are sharing ideas and information [16].

During this research, we defined the outline of a project to design, develop, and implement the software for an interactive, distributed, multimedia instructional system (MMIS) for distance education. Under this project, learning materials will be designed for interactive use by single individuals or groups of students. The aim of the project is to incorporate interactive multimedia technology into existing distance education methods and produce an interactive

instructional system to be used for distance education. The instructional material for MMIS will be designed using multiple media - text, graphics, audio, and video. The course material will be authored in hypertext. Thus the course material includes non-linear connections of traditional text with non-textual (image, sound, and video) resources. The model used for authoring the lectures will be the interactive timeline model (ITM).

The proposed system is interactive, as it incorporates interactive multimedia techniques, such as simulations, interactive tutorials, communication, and collaboration methods, into existing distance education methods to inspire active learning by letting students participate in the instructional process. This system is 'distributed' in the sense that information sources (data) are distributed over several databases and also because of the geographically dispersed nature of its users.

The following features are added for interaction in the MMIS system: an audio link between all users and instructor, video conferencing, email, whiteboard, and application sharing. Table 1.1 shows the various styles of interaction and modes in which interaction can take place. Table 1.1 indicates that

an authority controls or manages the interaction session. It can be the instructor, server, or one of the groups taking part in the interaction. Table 1.1 shows the scenarios in which this system would be used. A description of the fields in the table is as follows:

1. Mode - Mode indicates synchronous/asynchronous (S/A) mode of interaction.

Synchronous refers to human-human interaction, and Asynchronous refers to human-computer interaction.

2. Interaction Styles - The following interaction styles are possible :

Virtual Office Hours - One or many students interact with the instructor at a specified time. The mode of interaction is synchronous. In this style of interaction, students can use videoconferencing, audio link, and/or application sharing tools to communicate and collaborate.

Virtual Lecture - The instructor interacts with one or many groups of students. The mode of interaction is synchronous. A hypertext lecture is delivered to the students, with audio and video as additional features.

Work Group - Groups interact with each other, using whiteboard, application sharing, or audio. The mode of interaction is synchronous.

Prerecorded - The instructional material is prerecorded and stored on the Instructor's server. Hence the Server interacts with one or many groups. The mode of interaction is asynchronous.

Tutorial - A tutorial refers to an interactive tutorial, created by the instructor. This form of interaction may be synchronous or asynchronous. In the synchronous mode, the instructor directs the events in the tutorial. This tutorial can be a part of the Virtual Lecture described above.

3. Topology- Topology indicates the direction of information delivery.

4. Access - Access indicates the authority in control during an interaction session.

$I \Rightarrow G$ means that the instructor controls and manages the entry/exit of students from the interaction session. I = Instructor, U = User/Student, G = Group of Users, S = Server.

5. Media - Media indicates the multiple forms of media, or means used for an interaction. Multiple media are hypermedia HM (audio,movie, text,graphics), WB(white board), AS(application sharing), AU(audio), Real time V(Video).

Table 1.1 Interaction Styles in MMIS

Mode	Interaction	Topology	Access	Media
S	Virtual Office Hours	U => I G => I	I => U I => G	HM, WB, AS, AU, V
S	Virtual Lecture	I => G	I => G	HM, WB, V, AU
S	Workgroup	G => G	G => G	WB, AS, AU
A	Prerecorded	S => U S => G	S => U S => G	HM, AU, V
S	Tutorial	I => U	I => U	HM, AS, AU
A	Tutorial	S => U	S => U	HM, AU

The MMIS project is interested in increasing the level of interactivity in an instructional process, so that the remote student has the choice to drive the process, instead of being passively driven by it. Interactive modifications of the teaching system during use permits matching of the student's learning progress so as to maximize the benefit of the learning experience. Use of ITM as the model for MMIS will not only provide various perspectives in the teaching of a subject, it will allow the user to explore the subject as deeply as needed while the incorporation and utilization of various resources will keep the interest level high.

The rest of the chapters are organized as follows. Chapter 2 defines a document model, describes the various models of time, states the requirements for

interactive multimedia document models, and compares various models discussed in the current literature to the ITM model. Chapter 3 describes the layered ITM model, detailing the features of the model, its limitations and uses. In Chapter 4, we give a background on usability testing, state the objectives for evaluation of ITM, describe the evaluation methodology, and present the results of evaluation. Finally, we conclude with suggestions on improving the model by adding more interactivity to the model and using it to create interactive learning materials.

CHAPTER 2

2. MULTIMEDIA DOCUMENT MODELS

2.1 An Overview

In this chapter, we introduce document models for multimedia documents, and compare the current research on models for multimedia documents with the ITM model. In section 2.2, we give the definition of a document model, and describe the document models for traditional and multimedia documents. In section 2.3, we introduce the terminology used for developing the model for interactive multimedia documents, define logical, temporal, presentation and control structures for multimedia data, and emphasize the need for modeling these structures in multimedia documents. We also present the different views of a multimedia document. In section 2.4, we introduce the terminology for modeling time in a multimedia document. The terms described in this section are used in the current literature to model time and we introduce them here to compare how document models use different models of time to represent the temporal structure of a multimedia document. We state the requirements of the model for interactive multimedia documents in section 2.5, describe the models presented in literature

that fulfill some of these requirements, and compare our model for interactive multimedia documents with these models.

2.2 Document Models

2.2.1 What is a document model ?

Documents communicate information and are structured according to certain rules. Document structure can be expressed in terms of abstract objects, hierarchical links between these objects, ordered and unordered objects, and shared components. A directed graph organizes these concepts, allowing a document to be fully described in graphical form. A document model provides a method of completely specifying the structure of the contents of a document. The document model for traditional documents defines a structure for representing static media such as text and graphics. The document model for multimedia documents defines a structure for the representation of documents in terms of the organization of the information they contain and the synchronization of the presentation of that information [7].

2.2.2 Structuring traditional documents

Traditional documents are composed of text and geometric graphics. The ISO's Office Document Architecture (ODA) standard systematizes the representation of traditional documents frequently used in an office environment, such as reports, letters, forms, and memoranda as described in [7]. ODA is a model that provides a method for creating the structure of a traditional document. The cornerstones of this model are its logical and layout structures. Figure 2.1 provides an example of an ODA document.

The logical structure of a document organizes the document content sequentially, and is intended to correspond closely to the functional purpose of the document. Layout structure defines where the document contents are to appear on a surface, when displayed or printed, and organizes the content to aid understanding. In ODA, the logical structure organizes the contents into chapters, sections, and paragraphs. The layout structure organizes the content into blocks (areas within a page), frames (groups of blocks), and pages. The logical and layout structures are joined through the associated content portion.

The current ODA standard supports only static media types: text, raster graphics, and geometric graphics. The ODA standard has been used to represent

multimedia documents, as described by Kalfallah and Karmouch [7]. Continuous media such as audio and video cannot be incorporated into ODA-compliant documents. Continuous media execute in time, and thus have temporal properties. ODA does not represent the temporal relationships between the objects within a document. HyperODA is an extension to ODA for handling time-based multimedia objects as described in [7]. It is not clear if HyperODA can handle synchronization of multimedia objects presented. The ISO Multimedia and Hypermedia Expert Group (MHEG) has defined standards for representation and coding of multimedia hypermedia objects as described in [7]. The purpose of the MHEG standard is to facilitate the interchange of multimedia information through telecommunication networks. However, synchronization specification is not included.

2.2.3 Structuring multimedia documents

Multimedia documents differ from traditional documents, since they contain continuous media like audio, video, and computer generated graphics, along with text and geometric graphics. A passive multimedia document integrates continuous media, like audio, video, and computer generated graphics, into a traditional document, but it is represented in a static visual form, later

activated by the user. Active multimedia documents contain continuous media, and each media object is treated as an object to be presented in time. An interactive multimedia document is similar to an active multimedia document, but also allows the user to interact with the objects, when they are presented.

A multimedia document has both spatial and temporal properties, i.e., data objects in a multimedia document must be ordered in space, as well as time. This implies an ordering of what objects will be presented, in what manner they will be presented, and in what order they will be presented relative to each other or relative to a reference point in time. To represent an active multimedia document, we need a model that allows us to specify how the contents of the document (multimedia data) should be organized (logical structure), ordered in space and time (spatial and temporal structure), and how these contents will be presented to the user (presentation structure). We also need a control structure in interactive multimedia documents. By means of this structure, we can control the interactions of the user or other applications with the presentation of the document.

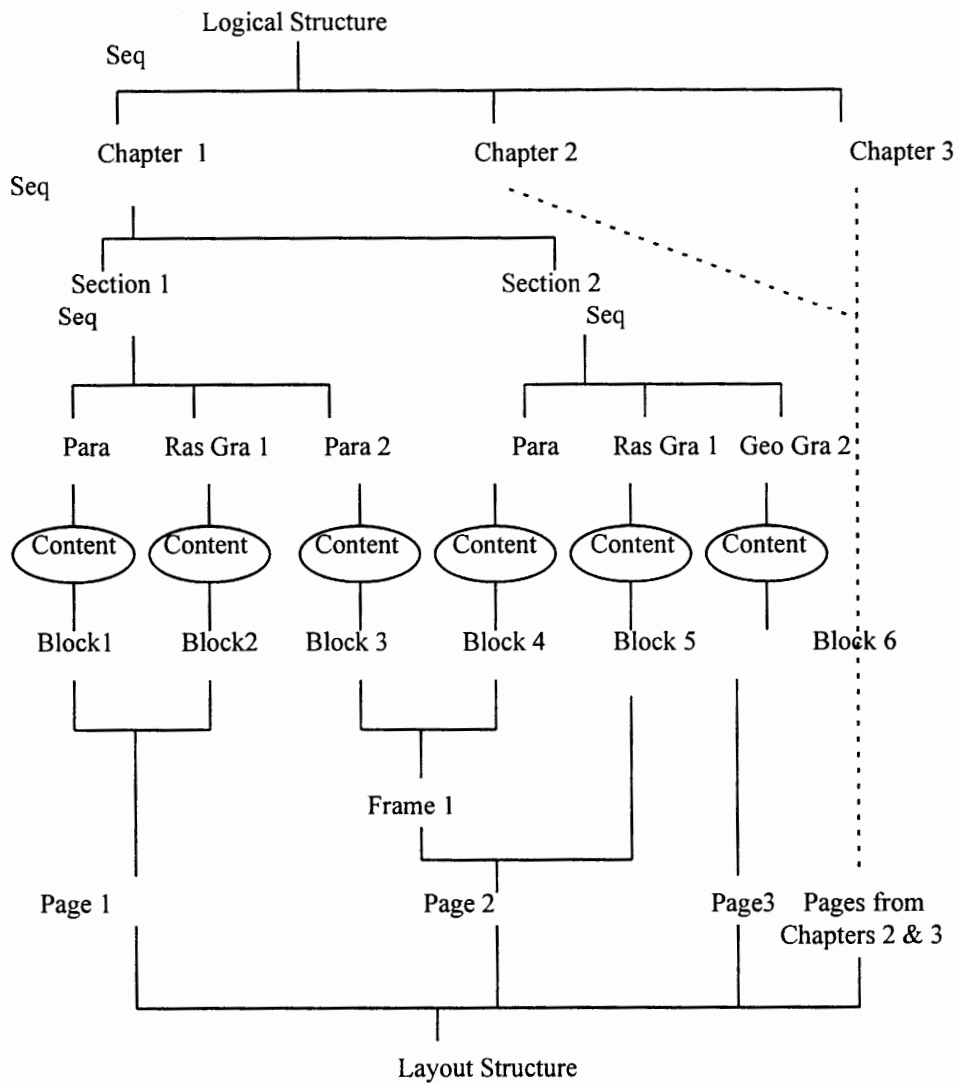


Figure 2.1 : Example of an Office Document Architecture structured document (Seq=Sequential, Para=Paragraph, Ras Gra=Raster Graphics, Geo Gra=Geometric graphics)

2.3 Modeling Multimedia Documents

2.3.1 Terminology

2.3.1.1 Multimedia Document Model

The terminology for methods to represent multimedia documents is not standardized in the current literature. Several research papers use the term "model" for a method to represent multimedia documents [2-6], others use the term "architecture" for the same[1]. We describe our interactive timeline model as used in a number of perspectives to show that it supports all these views for document creation. The authors may use one or more of these approaches for authoring. We define the various structures of a multimedia document, describe various views of a multimedia document from an author's perspective, explain how these views can represent the structure of a document and describe a model as a union of all these views.

2.3.1.2 Multimedia Scenario

Multimedia scenarios are fully specified temporal entities involving multiple media [2]. A multimedia scenario is a union of the spatial, temporal, and presentation structures of data objects in a multimedia document. When developing the design of a document, the author can start by describing the

scenario in words, and set up the structure of the document. The following sections describe the structure of a multimedia document.

2.3.1.3 Logical structure

The logical structure of a multimedia document specifies what data objects will be included in the document, and organizes the data according to the functional purpose of the document. Multimedia document authors must take into account both the content and the context while designing the logical structure of the document. The content includes issues like what material will be included, how it is structured, and how it is to be accessed. The context of the system is how it is to be used - whether it is an information retrieval system for education, training, or entertainment purposes.

2.3.1.4 Temporal Structure

In an active multimedia document presentation, media objects presented have an implied duration. For example, audio and video sequences are normally presented at the rate at which they were captured. Static media do not have implied duration. However, by assigning a duration to every object in the document, the author can create the temporal structure of the document. The temporal structure of a multimedia document specifies the ordering relationship

that determines when the data is to be presented relative to one another or relative to a reference point in time.

A multimedia document model should provide a separation between the data and its temporal structure. If the same data objects can be shared by several temporal structures, they need not be duplicated each time they are to be a part of a new structure. Avoiding duplication is a concern of all database systems, and it is particularly important in multimedia databases in which individual objects are storage intensive.

2.3.1.5 Presentation structure

The presentation structure specifies how multimedia data will be presented to the user in space (spatial component) and time (temporal component). The spatial component of the presentation structure specifies how the content of the document will appear to the user when it is presented on a display device, such as a monitor. It describes the spatial properties of media objects to be presented during playback. For example, the author can assign portions of the document's contents to physical locations on the display device.

The temporal component of presentation structure ensures that the media objects are presented only for the duration specified by their temporal structures.

It also synchronizes the playback of various objects. For example, the author may want that a video clip be accompanied by its audio track, followed by display of text accompanied by its audio track. The temporal component makes sure that the presentation of video and audio are synchronized, and that the display is cleared before text is displayed, and the next audio track is synchronized with the presentation of the text object.

It is important to distinguish the temporal structure of data from its presentation structure. The temporal structure is an ordered collection of data objects, and implies an ordering in time. A single temporal structure may have several presentation structures. For example, if the viewer varies the speed of playback of a continuous media object at the time of authoring, it represents different interpretations of the same temporal structure.

Visual media objects are different from audio objects because their contents must be physically laid out for presentation on a display device. The presentation structure also specifies the characteristics of playback of audio objects as well as visual media objects such as text, video etc. It specifies the

volume, playback speed, font type and size, window frame size, color, user interface components, such as buttons, scroll bars, menus etc.

2.3.1.6 Control structure

The control structure of an interactive multimedia document describes how the interaction of the user or other applications with the presentation will be controlled. It specifies the options provided to the user or other applications for interacting with the presentation, what actions are expected from the user, and what are the responses to those actions. It provides the interfaces that allow the user to interact by changing the variables in the presentation, for example, volume of an audio track, or direction of playback.

2.3.2 Different views of a multimedia document

We describe a multimedia document model as a union of the different views of a multimedia document (Table 2.1). These views are hierarchical as shown in Figure 2.2. The scenario view allows an author to qualitatively describe the logical, temporal, presentation, and control structures of the document. The specification view allows one to transform the scenario into a quantitative description of the structure of the document. The playback view allows the author

to present the document, and synchronize the presentation of the objects, and control the interaction of the user and other applications with the presentation.

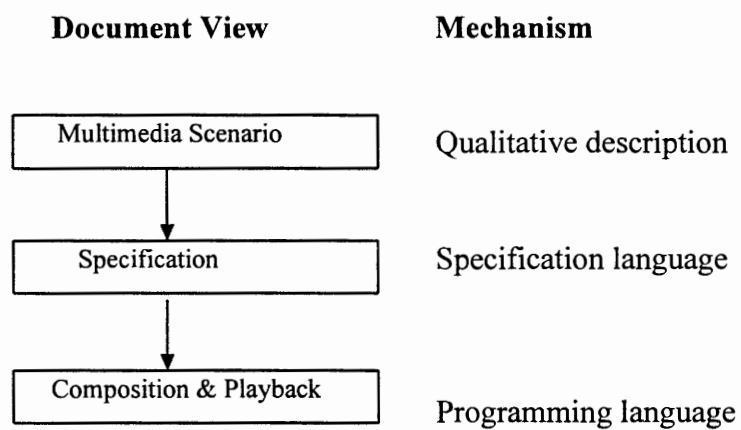


Figure 2.2 : A hierarchical representation of different views of an interactive multimedia document

Table 2.1 : Different views of a multimedia document

(L=Logical, T=Temporal, P=Presentation, C=Control, P/SC= Spatial Component of presentation structure, P/TC=Temporal Component of presentation structure, P/Other = Other characteristics of objects specified by presentation structure).

(a) Scenario View

Structure	Mechanism / Abstraction
L + T + P +C	A qualitative description in words.
Logical	What data objects will be included in the document - logical organization of objects according to function of the document, and which particular logical structure will be used for example, tree-structure, directed graph etc.
Temporal	A description of order of presentation of objects using a qualitative model of time (partial ordering in time).
Presentation	A description of user interface, playback devices used, spatial layout of objects on screen etc.
Control	A description of options provided for interaction, and how the interaction of the user or other applications with the presentation will be controlled.

(b) Specification View

Structure	Mechanism / Abstraction
L + T + C + P/SC , P/TC, P/Other)	A formal specification language used. Quantitative description of document structure.
Logical	Logical organization of objects according to a particular logical structure.
Temporal	Ordering, parallelizing of objects in absolute time. (complete ordering in time).
Presentation Spatial	Spatial layout of objects on screen, window coordinates, window size.
Temporal	Mechanism of ensuring that the objects are presented according to their temporal structures, mechanism of synchronization of objects during playback.
Other	Appearance of elements of the user interface, font type and size, format of text, color map, image resolution playback devices used, playback parameters, such as, speaker volume or playback speed (frames per second).
Control	Mechanism of interaction. Specification of options provided to the user (or other applications) for interacting with the presentation, what actions are expected from the user, and what are the responses to those actions.

(c) Composition and Playback View

Structure	Mechanism / Abstraction
L + T + P + C	Programming language for creation of interfaces for document creation and playback.
Logical	Interfaces for creation of the logical structure of the document according to the specification.
Temporal	Interfaces for creation of temporal structure of media objects.
Presentation	Interfaces for interpreting the logical and temporal structures, and executing the document or playback of document.
Control	Interfaces for interaction, for example, response to user actions. Interfaces that allow the user to change the variables in the presentation, for example, volume of an audio track, or direction of playback.

2.4 Modeling Time in a Multimedia Document

2.4.1 Events

Activities in a multimedia scenario are called "events". An event is an occurrence in time that can be instantaneous or can occur over some time period. For example, consider a scenario describing an instructor lecturing with slides. The time that each slides stays in the projector depends on the time expended in explanation. In this way, the activity called "slide in projector" is dependent on the activity called " instructor explains slide". These two activities are related as well as dependent in time. This relation or dependency must be captured by the model of time that represents this scenario.

2.4.2 Temporal Specification

The temporal structure of a multimedia scenario is translated into a temporal specification, which is a representation of the scenario that can be presented in the time domain. A formal specification language is used for this translation. A specification language is a combination of semantic and syntactic domains [3]. For a temporal specification language, the semantics is specified by

the temporal structure itself. The syntactic domain is defined in terms of symbols, and a set of grammatical rules.

2.4.3 Synchronous and Asynchronous Events

We borrow the following definitions from [4] to differentiate synchronous and asynchronous events. Synchronous events are events with predictable times of occurrence, that is whose temporal placement is known in advance. Asynchronous events are events with unpredictable times of occurrence, that is whose time of occurrence cannot be known in advance. Asynchronous events are studied, because in multimedia systems, there are two sources of indeterminacy: user interaction, in which the final realization is not known until the user interacts, and system failures, where the final realization is not known until run-time.

2.4.4 Temporal Equality

Temporal equality is a synchronization constraint requiring that two events either occur simultaneously or that one precedes the other by a fixed amount of time. Temporal inequality is a synchronization constraint requiring, for example, that events A, and B occur, so that A precedes B by an unspecified duration, by at least some fixed time, or by at least some fixed time, and at most another fixed time.

2.4.5 Models of Time

A temporal scenario is a set of independent events or a description that captures temporal relationship among the events. A temporal model captures the relation of events in a temporal scenario. A classification of models of time is given in [3]. The authors use three concepts to classify temporal models : the basic time unit, the contextual information associated with the basic time units, and the type of time representation technique. There are five general classes of models of time to which the authors have assigned the following names:

1. Qualitative Dates
2. Qualitative Instants
3. Qualitative Intervals
4. Quantitative Dates
5. Quantitative Intervals

The basic time unit is the temporal unit used in characterizing the temporal scenario. There are two basic time units, instants, and intervals. An instant is a zero length moment in time. An interval S is a set $\{x \mid a \leq x \leq b\}$, where a and b are two time instants. The contextual information specifies the type of information

that can be associated with instants or intervals of time. It is described as either qualitative or quantitative. Quantitative information is the temporal information that can be expressed using a real or virtual measure of time, for example seconds, or bits in a constant bit rate stream. Qualitative information is the temporal information that is not quantifiable. Qualitative information relating to basic time units can be of the following types:

1. Basic binary temporal relationships between instants - It defines three ways in which two instants can be related - before, after, and, at-the-same-time.
2. Basic binary temporal relationships between intervals - It defines thirteen ways in which two intervals can be related. For example, two intervals can 'start' at the same time, 'finish' at the same time, 'overlap' each other, one interval can be 'before' or 'after' another interval etc. Similarly, an n-ary temporal relationship between intervals defines the ways in which n intervals can be related in time.
3. Indefinite temporal relationships - These are temporal relationships between basic time units that are not explicitly stated. For example, { 'before' or 'at the same time' } , { 'at the same time' or 'after' } are indefinite temporal relationships.
4. Duration relationships - These relationships describe how the durations of two temporal intervals can be related. For example, [a,b] is shorter than [c,d] if the duration (b-a) is less than the duration (d-c).

Time representation techniques describe how time can be captured and mechanized in a computer environment. A particular representation occurs as a result of the application of a model of time.

2.5 Requirements of the model for interactive multimedia documents

In section 2.5.1, we state the various requirements of the model for interactive multimedia documents, describe models that fulfill some of these requirements, and compare ITM model with these models. In 2.5.2, we describe the model as a software product, define various desirable qualities of software, and emphasize that these qualities must be present in the model and its implementation.

2.5.1 Document structure requirements

2.5.1.1 Logical structure

The model must specify what data objects will be included in the document and provide a structure that organizes the data according to the functional purpose of the document. Structuring data for a multimedia document is time-consuming, and we need a document model that provides an efficient way to manage and reuse multimedia data once it has been created. Also, the model

should provide a separation between the data and its presentation. The presentation often involves elaborate specification of many variables like, layout, rate, color, UI components and so on. Hence, it is desirable that these specifications be reusable.

Multimedia provides a richness in data types that facilitates flexibility in expressing information. Hypertext is a control-structure that provides an elegant way of navigating through this data in a content based manner. Hypermedia is a combination of multimedia and hypertext systems, and allows us to organize information in accordance with the ways in which we naturally access and manipulate it. In hypertext systems, nodes, links and anchors are used to define a logical navigation mechanism in a document. Basic structuring techniques break the information into atomic blocks, commonly called nodes. Nodes contain an item of information like images, audio track, blocks of text, video, and so on. Nodes can be organized in a linear, hierarchical, or a network structure. Documents can use more than one logical structure.

The Dexter model [9] focuses on the basic node/link network structure that is the essence of hypertext. The fundamental entity and basic unit of

addressability in its storage layer is the component. A component is either an atom, a link, or a composite entity made from other components. Atomic components are what are typically thought of as nodes in a hypertext system. Links are entities that represent relations between other components.

The Amsterdam Hypermedia model (AHM) combines hypertext and multimedia and structures data objects into atomic and composite components [8]. The atomic component contains meta-information that refers to a particular data block, while the composite component defines such information for a collection of atomic or composite blocks. The AHM atomic components contain link anchor information, presentation information, and component attributes.

The ITM model uses several ways to organize information for the logical structure of the document. The text objects within a unit are sequential relative to each other, but there can be hypertext links, and embedded Tcl code (active text) within the text objects. Also, there can be links from one text object to another within a unit, links from a text object in one unit to another unit on the same timeline or links to a different timeline. The user is provided with several options for navigating the document, thus increasing the level of interaction with the presentation.

Woelk and colleagues developed an early object-oriented model for multimedia documents, as described in [6]. The model provides a hierarchy of object types oriented towards the aggregation of data. For example, a memo might consist of a header, a body, and a trailer; a body might consist of one or more paragraphs or drawings; a paragraph or a drawing object might be associated with the data it represents and the method through which it would be displayed. Despite the structured nature of the aggregation hierarchy, arbitrary links are allowed between objects to support the addition of hyperlinks. Data can be shared among documents, either by references to data or by data copying.

The traditional timeline model aligns all objects to be presented on a single horizontal axis representing time. On the vertical axis, several media objects can be aligned for simultaneous presentation. The objects are organized in a linear structure in this model.

Buchanan and Zellweger [4] propose the Firefly model to represent general multimedia scenarios. Each media object is modeled by two connected rectangular nodes representing start and end events. Though the Firefly model can

represent asynchronous events, it becomes hard to trace in complex interactive documents.

Little and Ghafoor [5] proposed the OCPN (Object Composition Petri Net), a model for description of general multimedia scenarios (Figure 2.3). The model uses a graphical description for the scenario, which is based on an extended type of Petri net. However, unlike the ITM model, the graphical nature of the petri net can become complex, and difficult to grasp when the document becomes relatively large.

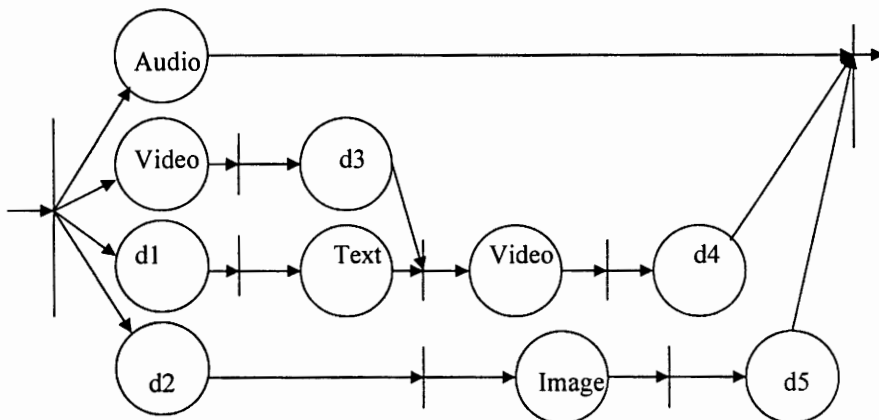


Figure 2.3 : How the Petri Net model would represent the scenario presented using traditional timeline (Figure 1.1)

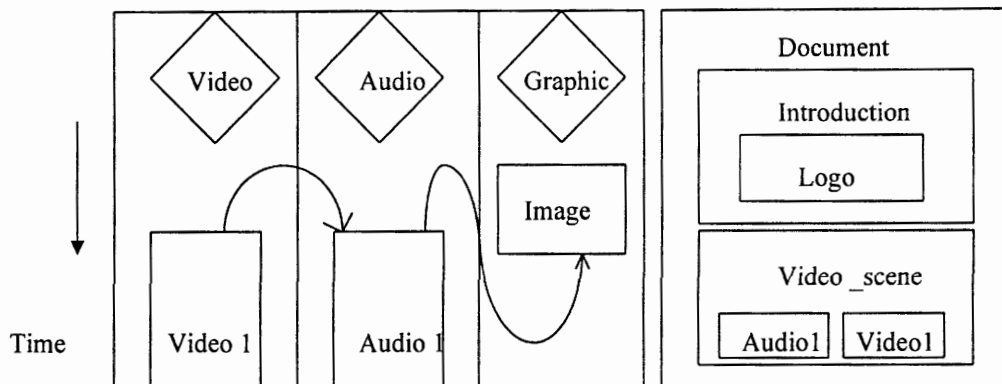


Figure 2.4 : CMIFed Channel view (left) showing sync-arcs to synchronize the start of the audio and video with the end of the logo and hierarchy view (right)

The CMIFed multimedia authorer provides a traditional timeline visualization called the "channelview" [10]. The hierarchy view offers a novel way of visualizing both the structure of the scenario, and the synchronization information using nested boxes. Vertically stacked boxes are executed in sequential order, while horizontally arranged boxes are executed in parallel (Figure 2.4).

In the relational grammars model, multimedia documents are presented automatically based on parsing and translation [11]. Grammar rules map the content to the look and feel of a spatially and temporally laid out document. This

model mainly addresses documents with alternate spatial layout, and unlike the ITM model, does not deal with user interaction with an active document.

A timeline model for active multimedia documents is described in [2]. The authors expanded the traditional timeline model to model active multimedia documents graphically within the timeline. This model introduces a new media type called "choice" that is placed directly on the timeline, and increases the power of the timeline. The new object is associated with a data structure that contains several fields like `user_action`, `region`, `destination_pointer` etc. When the scenario containing choices is presented, the user can make an action that initiates the choice. For example, the author can specify that if the user clicks on a region of the screen at a certain time, the presentation will "jump" to a new chapter. However, this model does not address the logical layout of an interactive multimedia document.

2.5.1.2 Temporal structure

The model must allow the author to specify the order in which the objects of a multimedia scenario should be displayed relative to each other or relative to a

common point in time. The model should allow the creation of the temporal structure for both synchronous and asynchronous events.

The timeline model is the most basic method used for temporal specification. It consists of a dated timeline, and all events are completely specified along a time axis. Hence, this model belongs to the category of quantitative dates. The contextual quantitative information corresponds to the exact date - "at" of the basic time unit. During presentation, the timeline can be interpreted , and various actions executed at the indicated moments in time. This model requires a total specification of all temporal relationships between media objects along a timeline, hence it can model all temporal relations excepts the ones with uncertainties. The ITM model is similar to the traditional timeline model, since it requires a total specification of all temporal relationships between objects. However, the ITM model is flexible enough to handle interaction during a presentation. In the timeline model for active multimedia documents [2], temporal inequalities between events can be modeled. However, the user cannot interact with an asynchronous event once it has been activated.

In the Firefly model, the contextual information corresponds to a binary temporal relationship between two instants. Each media object is modeled by two

connected rectangular nodes representing start and end events. Any other event used for synchronization is represented by a circular node placed between start and end events. Asynchronous events contained in a media item are represented by a circular node that floats above the start event. Temporal inequalities between events are represented by labeled edges connecting these events. This model cannot represent the temporal relationships between activation and deactivation of asynchronous events with other events. The ITM model is simple and graphical, and can easily represent complex temporal scenarios. The user can deactivate an asynchronous event at any point in the presentation of a timeline.

In the OCPN model, the contextual information is both qualitative and quantitative. Temporal relationships are fixed, hence no indeterminacy can be expressed in the temporal scenario. This model accommodates temporal inequalities, such as when a delay is unknown at authoring time, thus allowing user interactions. However, unlike the ITM model, the graphical nature of the petri net can become complex, and difficult to grasp when the document becomes relatively large.

Hoepner defines a temporal specification scheme for description of general multimedia scenarios, as described by Little and Perez-Luque [3]. This scheme

consists of a set of path operators with an associated graphical representation. These operators are valid for any synchronization mechanism that can interpret them. The model of time used, belongs to the general category of qualitative intervals. The contextual information corresponds to a subset of basic binary relationships, and to a subset of indefinite temporal relationship between intervals. In this scheme, the path operators limit the number of binary temporal relationships that can be modeled. Indeterminacy in temporal relationships can be modeled.

The ITM model uses the n-ary temporal relationship between intervals, and the duration relationship scheme. The n-ary temporal relationship between intervals is used for objects that are of different media types, and the duration relationship scheme is used for objects of the same media type within a unit. This scheme does not limit the number of relationships that can be modeled for objects of the same type. However, for multiple instances of the same type of object, the number of duration relationships is limited by the length of the unit.

Wahl and Rothermel have proposed a temporal specification scheme that has a common set of operators that describe the temporal relationships between intervals and the possible variations due to user interactions. The model of time

used belongs to the category of qualitative intervals. The contextual information is qualitative and can optionally use quantitative information. Indeterminacy of interval duration in the temporal scenario can be expressed in addition to indefinite temporal relationships between pairs of intervals. This model has powerful operators that include both temporal equalities and inequalities between events. But, scenario representations are hard to trace in a complex interactive document. In addition, although operators can be applied to asynchronous events, they cannot be applied to the events that might not occur at all during the actual presentation.

2.5.1.3 Presentation structure

The model must allow the author to create the presentation structure (both spatial and temporal components) of the document. This structure specifies how multimedia data will be presented to the user in space and time. It should provide mechanisms for synchronization during the playback of various objects.

The model should specify the mechanisms to synchronize the simultaneous presentation of these objects. [1] suggests that an effective "rendering conductor" design is essential to execute scenarios and playback documents. In the Mediadoc architecture for multimedia documents proposed by

Karmouch and Emery [1], "rendering conductor" is the module that executes the scenarios. In the ITM model, the author can play back the document using the interfaces provided with the playback tool.

2.5.1.4 Control Structure

The model should support interaction of the user as well as other applications with the presentation. It should represent the actions associated with responses to user actions, as a part of its control structure. It should be flexible enough to allow the user to take control of the presentation, and change the course of the presentation, and regain control if the user takes an action that is not expected. The model should be able to provide the playback tool with runtime information such as user interaction, and also pass information between other applications and the tool.

In the model for active multimedia scenarios [2], the user interaction with the presentation is limited. Every choice object has a duration, and the user has a window of opportunity to make an action that initiates a choice. If the user does not take any action within that time, he loses the chance to make it. Also, the user has no interaction with the presentation, once an asynchronous event becomes activated. It is deactivated only when the presentation flows to another timeline.

The ITM model uses this model as a starting point, and extends and modifies this model to meet the requirements of an interactive multimedia documents.

2.5.2 The Model as a Software Product

The model for a multimedia document is a software system that will be delivered to the authors (users of the model), who can create interactive multimedia documents for viewers (viewers of the document that the author produces). Thus, the model is a software product, and it is required to have some, if not all, desirable qualities of software. Some of these apply both to the model and to the process used to produce the model.

From point of view of the developer of the model, it should be verifiable, maintainable, portable, and extensible. From the authors' perspective, the model should be reliable, efficient, and easy to use. From the viewers' point of view, it should be robust, user-friendly, portable, and configurable. In this section, we define these software qualities, and state why it is important to incorporate these in the model. We borrow the definitions of important qualities of software products and processes from [12].

1. Correctness, Reliability, and Robustness - The terms correctness, reliability, and robustness are often used interchangeably to characterize a quality of software that implies that the application performs its functions as expected. A clarification of the terminology is needed to better understand and analyze the underlying issues.

Correctness - Correctness is a mathematical property that establishes the equivalence between the software and its specification. A program is functionally correct if it behaves according to the functional requirements specifications. If we are rigorous in specifying the functional requirements of the model, we can be more systematic in assessing its correctness.

Reliability - Software is reliable if the user can depend on it. The notion of reliability is relative, while correctness is an absolute quality. Any deviation from the requirements makes the system incorrect, whereas, if the consequence of a software error is not serious, incorrect software may still be reliable. A systematic design approach followed by a rigorous and disciplined implementation is the best way of building reliability into the

model. We should not trust the model unconditionally. We must be aware of the complexity of the model and critical areas where it can be unreliable. We must do all we can to minimize the risk of errors.

Robustness - A program is robust if it behaves "reasonably", even in circumstances that were not anticipated in the requirements specification - for example, when it encounters incorrect input data, or some hardware malfunction, say, a disk crash. If we could state precisely what we should do to make our model robust, we would be able to specify its reasonable behavior completely. Thus robustness of the model would become equivalent to its correctness. If we put a requirement in the specification, its accomplishment becomes an issue of correctness; if we leave it out of the specification, it may become an issue of robustness.

2. Performance - Performance of software can be equated with efficiency. A software system is efficient if it uses computing sources economically. Performance is important because it affects the usability of the system. It is required that the performance of the model be acceptable to the users. If the model is too slow, it would affect the productivity of the authors, possibly to the point of not meeting their needs. If it uses too much memory, it may affect other

applications that are run on the same system. If it uses too much disk space, it is too expensive to run. Performance also affects the scalability of the model. We must take performance into account while making design decisions for the model, for example, will the model be slower as the number of inputs and variables increases or the length of the program being processed increases.

3. User Friendliness - A software system is user friendly if its human users find it easy to use. User friendliness is a subjective term. If the user is a novice, a window interface and a mouse is friendlier than a set of commands. If the user is an experienced programmer, he might prefer a set of commands rather than a fancy window interface, to execute a task. The user interface is an important component of user friendliness. It is important that the model for interactive multimedia documents be user friendly. The user friendliness of the model is affected by the correctness and performance of the model. For example, if the model provides wrong answers, it is not user friendly, regardless of how good the user interface is. Also, if the model provides answers more slowly than the user requires, it is not friendly, even if the answers are correct. The multimedia authoring environment is more complicated than that for traditional documents due to the inclusion of temporal specifications. Special graphical user interfaces

are required to support the needs of authors and users during authoring and playback of multimedia documents.

4. Verifiability - A software system is verifiable if its properties can be verified easily. We would be interested in verifying the correctness and performance of the model. Modular design, disciplined coding practices, and the use of an appropriate programming language all contribute to verifiability.

5. Maintainability - Software maintenance is used to refer to the modifications that are made to a software system after its initial release. Maintenance can be divided into three categories - corrective, adaptive, and perfective maintenance. Corrective maintenance refers to the removal of residual errors present in the product, as well as errors introduced into the product during its maintenance. Adaptive and perfective maintenance are the real sources of change in software. Adaptive maintenance involves adjusting the application to changes in the environment, for example, a new release of the hardware or the operating system, or a new database system. Perfective maintenance involves changing the software to change some of its qualities. Maintainability can be viewed as two separate qualities: repairability, and evolvability. Repairability allows correction of defects

in software with a limited amount of work. Evolvability allows one to modify software over time, to provide new functions, or to change existing functions. The model must be maintainable, and to achieve this goal, it must have well-designed modules that are easier to analyze, and repair than a monolithic piece of code. Also, it must have the right module structure with the right module interfaces to reduce the need for module interconnections. The right modularization promotes repairability by allowing errors to be confined to a few modules, and making it easier to locate and remove them. Several modularization techniques, including data abstraction, can be used to make the model more maintainable.

6. Reusability - Reusability is a software quality that is strongly affected by anticipation of change. A component is reusable if it is directly usable to produce a new product. Thus, reusability can be viewed as evolvability at the component level. The model can be designed such that likely changes that we anticipate in the requirements, or modifications that are planned as part of the design strategy, may be incorporated in the model smoothly and safely. If we anticipate changes in a component of the model, then the component should be designed so that the changes may be accommodated easily. The layered multimedia data model (LMDM) is an example of a model that stresses the conceptual separation of

different parts of a multimedia system [6]. This leads to less coupled components, which are thus more reusable, more portable across systems, and more interoperable with one another.

The use of object oriented design unifies the qualities of evolvability and reusability. Mediastore is an object-oriented database that allows storage of multimedia documents created under Mediadoc architecture [1]. The database's object-oriented capability allows authors to easily define object classes such as chapters by subclassing other classes and reducing the time required to create object types, by reusing existing classes.

7. Portability - Software is portable if it can run in different environments. The term "environment" refers to a particular hardware platform or a software environment such as a particular operating system. It is desirable that the model be portable to different operating systems or hardware platforms. We need to use techniques that allow the model to determine the capabilities of the hardware and to adapt to them.

8. Understandability - The model for an interactive multimedia document can be complex. However, certain guidelines can be followed to produce more

understandable design. Understandability is a software quality that helps in achieving many other qualities such as evolvability and verifiability. The authors should be able to understand the model to create interactive documents that are clearly and concisely structured, and easily interpreted, understood, and modified. Also, the task of presenting a document should not be unnecessarily complicated.

9. Configuration management - Configuration management is the discipline of coordinating software development and controlling the change of software products and components. The model should allow the authors and users to manage the various versions and revisions of software in a controlled manner. The model must be kept consistent even when changes are applied to some of its components. It must be possible to store and retrieve documentation, source modules, etc., from a database that acts as a central repository of reusable components. Besides this, the model should allow the author and the user to configure the system according to their needs and the capabilities of the hardware or software system at hand. For example, the author and user can change the configuration of the playback devices such as display device, devices for audio playback etc., provided by the model and change it to what is available to them.

The model should be made easily adaptable by providing all configuration data in a module that can be accessed by the authors and users.

CHAPTER 3

3. THE INTERACTIVE TIMELINE MODEL

3.1 An Overview

This chapter explains the Interactive Timeline Model (ITM) in detail. Section 3.2 describes the design goals for ITM. The layered ITM model, and the document structure supported by ITM is described in section 3.3. Section 3.4 lists the interfaces provided by ITM. The features of ITM are given in section 3.5. Section 3.6 lists the limitations of ITM.

3.2 Design Goals

We had the following goals throughout the design and implementation of ITM :

1. To design a model that can support the creation of multimedia document structures.

To present interactive multimedia documents, we require a model that allows creation, storage, and presentation of active multimedia data in a clear and concise manner, yet is powerful enough to describe multimedia documents to the

extent required by the authors. Our aim was to create a model that provides the facilities and operations multimedia authors expect, as well as those necessary to permit and ease the process of creation and presentation of an interactive multimedia document. The author of an interactive multimedia document would expect the model to provide features that allow him to logically structure the contents of the document so that navigation through the document becomes easy, to specify the temporal and synchronization relationships among the media objects presented, to present the document using a playback tool that provides interaction with the user, and to control the presentation and interaction with the user. Thus, the design of the model should include the design of the logical, temporal, presentation, and control structures of the document.

2. To design a model that is functionally correct, robust, understandable, efficient, reliable, maintainable, configurable, portable, user friendly, verifiable, and has reusable components.

The design of the model should be such that it is easily understood by the author. It should be reliable, efficient, portable, and verifiable. The design of the model should be modular, so that it is easily maintainable. Creation of data

objects for a multimedia document can be time-consuming. Our aim was to create a model that allows reuse of data objects once they have been created. Also, authors and users should be able to configure some parameters in the model according to their needs. The present models of interactive multimedia documents lack features that allow the user to interact with the running presentation. One of our goals was to include features that let the user and other applications interact with the presentation, and also allow the playback tool to control the interaction. To present the documents, we included a playback tool in our design. To allow the user to readily interact with the presentation, we included the design of a user interface for the playback tool. Our interface decisions were motivated by the aim to develop a system whose use matches the goals, needs, and expectations of the intended users (both authors and viewers).

3.3 The Interactive Timeline Model

3.3.1 An Overview

We extended the model described in [2] to include features that increase the power of the model by increasing the level of user interaction with the multimedia document presentation. We present the interactive timeline model for

interactive multimedia scenarios. This model establishes the relationship between the logical, temporal, presentation, and control structures of an interactive multimedia document. It emphasizes the separation of multimedia data from its temporal structure, and its presentation structure. We implemented the features of the model using the scripting language, Tcl/Tk. In the following section, we present the layered ITM model, and the document structure supported by ITM.

3.3.2 The layered ITM model

ITM is a layered model as shown in Figure 3.1. The layered approach leads to clearly separating data management, composition, presentation management, and user interaction. The following sections describe the different layers of ITM in detail.

3.3.2.1 Storage layer

This layer defines the data format and semantics for generation of media objects for the multimedia scenarios. It allows the specification of data, and provides abstractions of data objects for the higher layers. It provides interfaces for creation of choice media objects. Multimedia data objects supported by the ITM model include persistent data (data that is already stored in a database), and

instructions to generate data. Non-persistent data, which is data whose duration is not known before it is presented, is not supported by ITM. An example of non-persistent data is the data that is accessed at run-time from a remote database. Using the interfaces provided by this layer, the author can create media objects such as text files text files with hypertext type links, or text files with embedded Tcl code, script objects such as Tcl script files, or choice objects like dialog boxes, message boxes, etc.

This layer also provides interfaces to add new procedures to the existing ones. We call the process of adding new interfaces a "registration mechanism". It is a mechanism by which the author can create and integrate custom objects, like dialog boxes or message boxes, new procedures, and events with the playback tool. Hence, the playback tool has information about the "registered" procedures. When the author registers a procedure, its name is appended to a list of default procedures provided by the model, and a numeric code is associated with the name of the registered procedure. The author can access this numeric code to execute the procedure. The presentation and run-time layer may use this code to control the execution of the procedure without the intervention of the author.

3.3.2.2 Composition layer

This layer allows the author to compose a multimedia document. It specifies the logical structure of the contents of the document and provides interfaces for grouping the data objects defined in the data definition layer into a multimedia scenario. The author can use these interfaces to create several scenarios, which can be represented by units on a timeline. Several such timelines can be created and linked with each other. The author can compose a unit by specifying what media objects would be included in the unit, or compose a timeline by specifying what units would be included in it. For example, the author may want to present a multimedia scenario using two timelines. He may want to compose the first timeline with two units and the second timeline by including a single unit. He may want to include a text file with an audio file in the first unit, and text, audio, and a choice object in the second unit, and put the two units on the first timeline. The author may include a script object, a choice object, and an audio file in a unit, and include it in the second timeline.

3.3.2.3 Presentation layer

This layer provides a description of when and how the data is to be presented on the display device and how the media objects will be synchronized during presentation. It provides interfaces to specify the spatial structure of the

data objects. Using these interfaces the author can specify where a media object is to be displayed on the display device. It allows authors to specify the attributes of media objects presented. In the example for composition of a multimedia scenario stated above, the author may want to play the audio file before the presentation of the text file in the first unit of the first timeline, display the choice object of the second unit at specific coordinates on the display device, or present the text in a particular font type or size. This layer also defines the user interface for the playback tool, and specifies how the user can traverse a document using links.

3.3.2.4 Run-time layer

This layer specifies the interfaces to start and control the flow of the presentation. It provides interfaces for sequencing and looping of timelines. This layer also describes the possible interactions between a user or an application and the presentation at run-time. It specifies the actions to be taken when a user or an application interacts with the presentation.

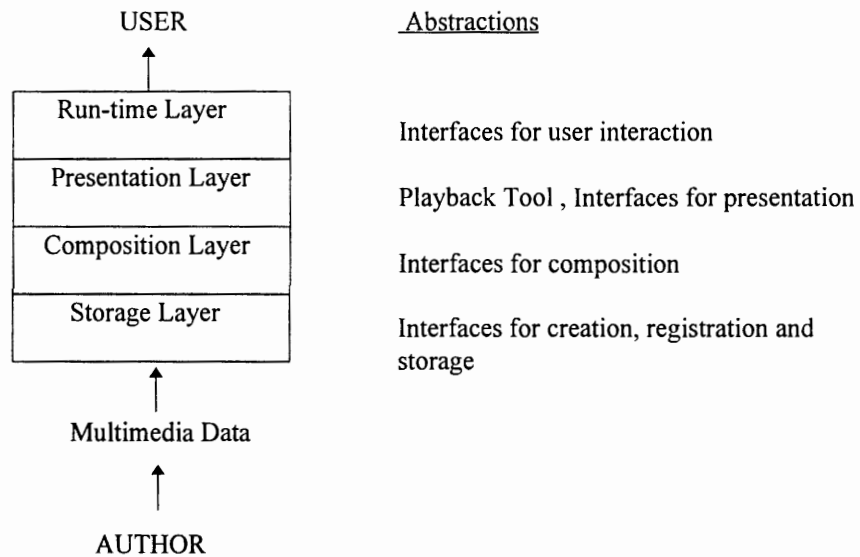


Figure 3.1 : The layered ITM model

3.3.3 Document structure

3.3.3.1 Logical Structure

The logical structure supported by ITM is hierarchical in nature. A unit is the basic component of the logical structure that can be created and presented on a timeline. Several media objects can be presented concurrently in a unit. Several units form a timeline and several timelines form the logical structure of the entire multimedia document. The different levels of abstraction in the logical structure are shown in Figure 3.2.

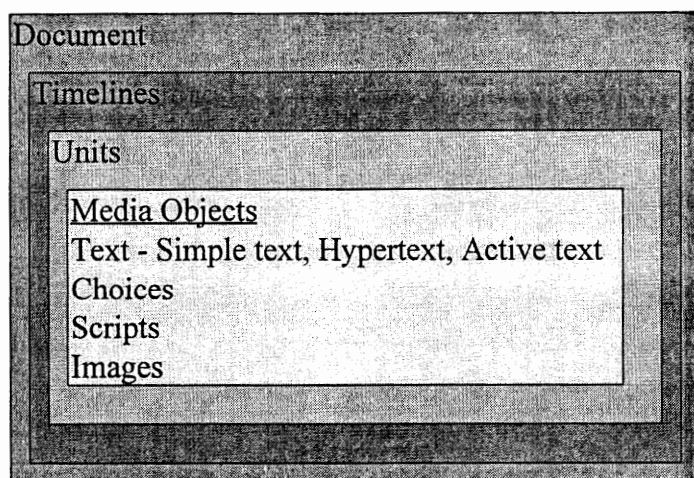


Figure 3.2 : Logical structure abstractions

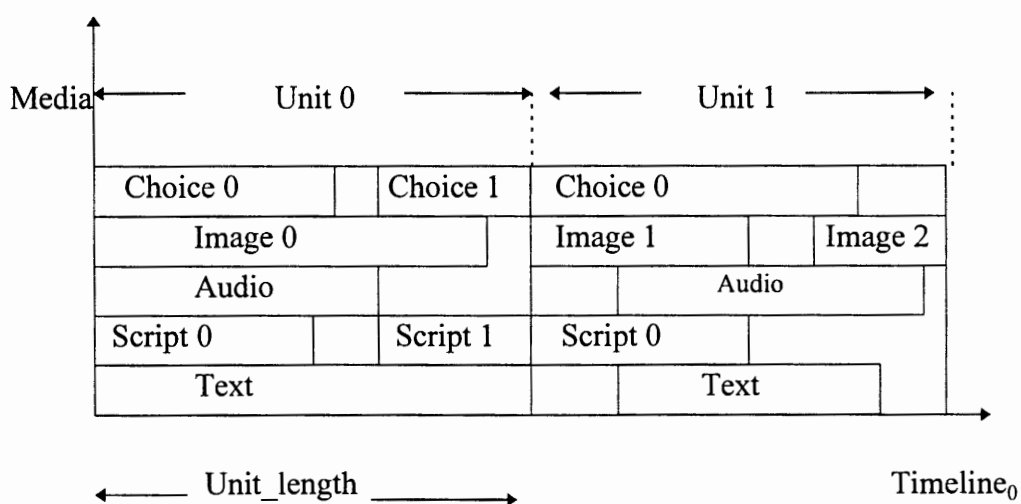


Figure 3.3 : Timeline representation of ITM

A description of the elements of the logical structure supported by ITM follows:

1. Timeline - A timeline is an alignment of events on a single axis representing time. In the timeline model, all temporal relations between media objects are fully specified in the order in which they should be presented. ITM uses multiple timelines to model the logical structure of multimedia scenarios. Each timeline is divided into a number of units. Figure 3.3 shows the timeline representation of ITM.

2. Unit - A unit on the timeline represents a part of a multimedia scenario. Each unit has an arbitrary duration, called length, and represents a multimedia scenario that consists of the media objects - text, images, audio, multiple scripts, and multiple choices. Units are required to support the ITM model, as the length is used to synchronize the presentation of the objects within a unit. Figure 3.3 shows two units on a timeline.

3. Media objects - Media objects supported by ITM are - Text, audio, script, choice, and image. A text object is a text file that may contain hypertext type links to other parts of text within the same text object, or it may contain "active text", which is a portion of text with embedded TCL code. Using the hypertext links in

text the user can jump to a different part of the text in a unit or jump to a text object in a different unit. The author can highlight the embedded code in the text and when the user clicks on the highlighted text, it causes a Tcl script to be executed. Using embedded code or "active text", the author may display information that allows the user to interact with the presentation. Audio and image objects play audio files and display image files respectively. A script object executes a Tcl script file. Choice is a special media object, and is described in detail below.

4. Choice - Choice is a new media object that increases the power of the timeline. Choice is placed directly on the timeline and presents the user with different options to change the course of the presentation. For example, when a choice is presented, the user can make a selection to jump to another timeline. There are two types of choices:

Implicit choice : In the form of buttons on the user interface - Back, Forward, Index, Demo, Help etc. At the end of each unit, the next unit is presented without intervention from the user. Also, at the end of a timeline, the next timeline is presented. This "default timeout" is also an implicit choice.

Explicit choice: It is specified by the author as a parameter of the choice interface. For example, the author may present a dialog box or message box to prompt the user to generate an event. When the user generates the correct event in the given time, the playback tool executes a procedure in response to the event. Using the choice interface, the author can associate an event with the response procedure. The author can register the response procedure using the registration mechanism. Thus, the playback tool can verify if the user generated the correct event, and execute the procedure associated with the event.

3.3.3.2 Temporal Structure

ITM uses duration relationships and n-ary binary relationships between intervals for the temporal structure of multimedia scenarios. The n-ary binary temporal relationships between intervals define ways in which n intervals can be related in time and are used to arrange text, audio, choice and script objects relative to each other within the unit. Duration relationships describe how durations of two temporal intervals can be related, and are used to model multiple objects of the same type within a unit, for example, multiple scripts and multiple choice objects. As shown in Figure 3.5, there are three levels of temporal

structuring in ITM, within a unit (intra-unit), between units (inter-unit), and between timelines (inter-timeline).

Inter-timeline - Temporal structuring of timelines in a document

Interval relationships (ir) - "Before", "After" among timelines - T_0 to T_n . For example, the timelines can be ordered so that the document begins with the timeline 0, and this is followed by timelines in increasing order of timeline numbers.

Inter-unit - Temporal structuring of units on a timeline.

Interval relationships (ir) - "Before", "After" between units - Unit 0 to Unit n. As shown in figure 3.5, units in timeline 0 are ordered in increasing order of unit numbers.

Intra-unit - Temporal structuring of media objects within a unit.

Interval relationships (ir) - "Before", "After" between media objects of type audio, text, script, choice, and image.

Duration relationships (dr) - "Before", "After" among multiple script objects, among multiple image objects, and among multiple choice objects.

A description of the terms used for temporal structure follows:

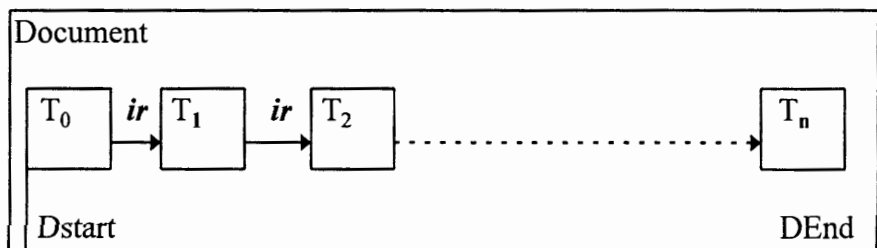
1. Start Time - Start time is the time at which an object is to be presented, relative to the start of the unit (Figure 3.5).

2. End Time - End time is the time at which the presentation of an object is to be stopped, and the object to be removed from the display. This time is also relative to the start of a unit.

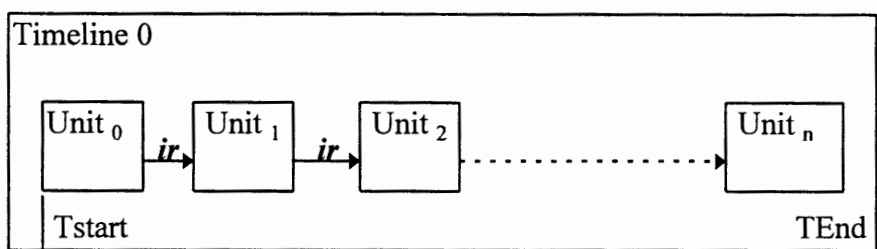
3. Duration - It implies the total time an object remains on the display device. It is the difference between end time and start time of an object.

4. Unit length - Each unit has a fixed, but arbitrary unit length, which is the duration of the unit. All media objects must be presented within this duration. All units may have different unit lengths. Unit length also serves as the synchronization point for presenting multiple units on a timeline. When the current unit is being presented, the next unit can start only after a time equal to the length of the current unit.

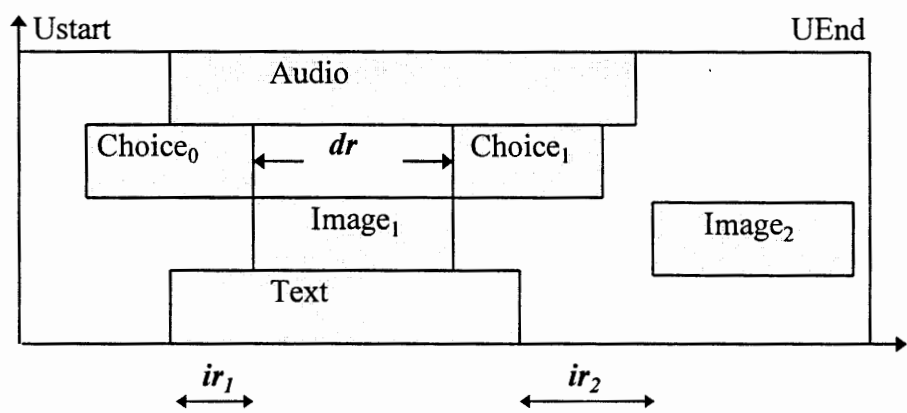
Inter-timeline structure



Inter-unit structure



UNIT 0 Intra-unit structure



T_0 = Timeline 0 dr = duration relationship ir = interval relationship
 $Dstart$ = Start of document $DEnd$ = End of document $Tstart$ = Start of timeline
 $Tend$ = End of timeline $Ustart$ = Start of unit $Uend$ = End of unit
 ir_1 = interval relationship (example - between choice₀ and text objects).
 ir_2 = interval relationship (example - between text and image objects).

Figure 3.4 : Temporal structure supported by ITM

3.3.3.3 Presentation Structure

Presentation structure specifies the spatial and temporal components of the presentation structure of data objects. The spatial component allows the author to describe the spatial properties of data objects eg., logical window name, window coordinates, window size etc. The temporal component provides interfaces that ensure that the objects are displayed only for their duration and interfaces for synchronization of units. It also describes the user interface for the playback tool.

The presentation structure also specifies how the user can traverse the document . Two directions for timeline traversal are specified. By default, the timeline is presented along the "x-axis" i.e., horizontally. The user can use the "implicit choice" - Back and Forward buttons on the user interface for the playback tool, to traverse the timeline horizontally. When the user makes a choice and jumps to another timeline, it is defined as the vertical direction in traversal of timelines. If the user fails to make a choice, the timeline continues to play along its x-axis. All unit links are in this direction. Vertical implies jumping from the current timeline to another timeline. When the user makes a choice, he traverses the graph of timelines from one timeline to another. All timeline links are in this direction.

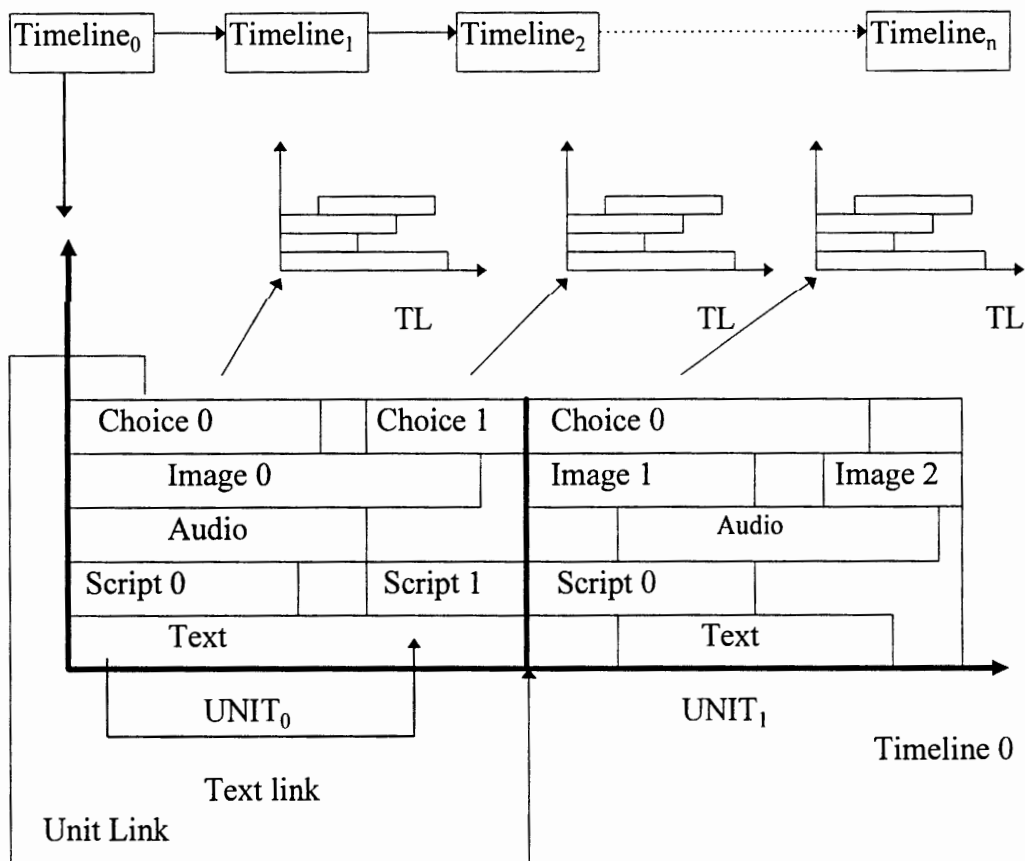


Figure 3.5 : Various links supported by ITM

The User Interface

The model provides an X-windows interface for the playback tool. The interface has been implemented using the Tcl scripting language [17]. It lets the author present the document, allows the user to interact with the presentation, and allows the tool to control the interaction. Figure 3.6 shows the user interface for

the playback tool. It has a text widget, in which text objects can be displayed. The user can resize, move, or scroll through text in the text widget. It has a frame at the top with buttons, which are used for implicit choice. The function of the buttons is described in Table 3.1. The model allows the author to change the configuration and function of the widgets on the user interface. Thus, the author has the flexibility of disabling a button on the user interface, changing the properties of the text widget, or adding more elements to the user interface.

Table 3.1 : Function of buttons on the user interface

User Interface Button	Function
Back	Displays the previous unit.
Forward	Displays the next unit.
Index	Displays the index of the document.
Home	Go back to start of the first timeline.
Pause	Pause the presentation. Click again to resume.
Demo	Starts playback of events pertaining to a particular choice.
Record	Start recording events.
Stoprec	Stops recording events.
Replay	Plays back the events recorded using Record button.
Help	Displays help on use of the playback tool.
Exit	Exits playback tool.

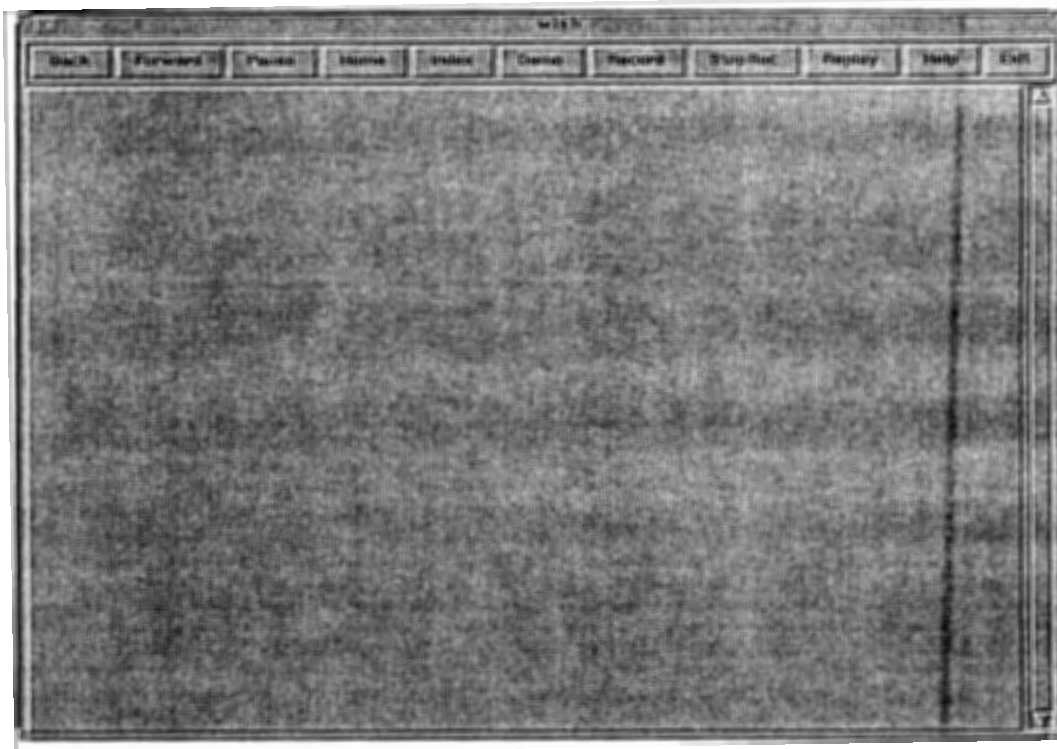


Figure 3.6 : The user interface

3.3.3.4 Control Structure

The control structure starts the presentation and controls the flow of the presentation. It provides interfaces for interpreting the logical and temporal structures of the document and executing or presenting the document. It provides a description of the types of interaction of the user and other applications with the presentation, and specifies how the interaction will be controlled. It specifies what options are available to the user for interaction, what actions are expected from

the user, and the responses of the playback tool to those actions. It provides interfaces that allow the user to change the variables in the presentation, eg., volume of an audio track or the direction of playback. It also provides interfaces that allow the playback tool to respond to user actions.

Implicit Choices - The playback tool provides the user with implicit choice objects, in the form of buttons on the user interface. The user can choose these choices and interact with the presentation at any point in time.

Interaction with user - With the help of default or custom choice objects, like dialog boxes, message boxes, and TCL scripts, the author can inform the user that a choice is available for a certain duration and the action expected from the user for making that choice. If the user responds with the expected action within the given duration, the playback tool responds to the user according to the choice presented. If the user makes an action that is not expected, the tool informs the user again about the expected action. If the user does not respond within the given time, either the tool executes the response, or the presentation continues along the original timeline.

Interaction with other applications - The model provides interfaces that allow other X-applications to interact with the presentation. The author can associate events related to other applications to procedures in the playback tool by registration of events and response procedures with the tool.

3.4 Interfaces of ITM

Table 3.2 categorizes the interfaces according to author-interfaces and support interfaces, and describes the purpose of different interfaces provided by ITM. The author-interfaces are the procedures available to the author to create and present documents. Support interfaces are procedures that are not visible to the author.

Table 3.2 Author and Support interfaces provided by ITM

(a) Author-interfaces

ITM layer	Name	Purpose
Storage	textobject	Creates text object
	audio	Creates audio object
	script	Creates a script object
	image	Creates an image object
	register_proc	Registers a new procedure
	Composition	unit
	timeline	Organizes units into a timeline
Presentation	ini	Initiates presentation. Starts background processing for control of interaction.
	play_chapter	Presents the timeline
Interaction & Control	go_back	Displays previous unit.
	go_forward	Displays next unit.
	go_home	Starts presentation from first timeline.
	index	Displays index of document
	demo	Presents actions expected from user for a choice.
	pause	Pauses and resumes presentation
	help	Displays help on playback tool
	exit	Exits playback tool

(b) Support-interfaces

Composition	set_default_codes	Enters the default codes for procedures in a list.
	display_codes	Displays codes for all procedures
Presentation	get_chapter_info	Gets information about the timeline - number of units in timeline etc.
	sched_next_unit	Schedules execution of the next unit.
	play_unit	Interprets the unit definition file
	sched_text	Schedules execution of text
	sched_audio	Schedules execution of audio
	sched_choice	Schedules execution of choice
	sched_script	Schedules execution of script
	sched_scr	Executes script
	parseFile	Displays active text in text widget
	loadFile	Displays text in text widget
	forAllmatches	Gets the first and last index of a pattern (to detect active text) in a text widget.
Interaction & Control	check_button_state	Enables/Disables Back and Forward buttons.
	display_choice_info	Informs user about a choice
	event_response	Looks for expected action and responds to user
	kill_choice	Destroys choice objects
	kill_script	Destroys script objects
	kill_audio	Cancels the execution of audio file.
	kill_image	Stops display of image

The author can create a document by defining objects of individual units using a unit-definition file. An example that shows how the author would create a unit-definition file is given in Figure 3.7. For format and syntax of the unit definition file, refer to the ITM Users Manual [18].

```
##### Timeline 0 #####
name Chapter 0
choice .msg 0 dialog=dproc1 1 4
# choice .msg 0 dialog=dproc5 5 15
script $script_path tcl1 1 5
textobject textdir Text00 1 10
audio audiodir dance.au 7 9
image imagedir yinyang.gif 10 15
unit_length 15
unit
name Chapter 1
choice .msg 0 dialog=dproc2 1 5
image imagedir smile.gif 3 8
script $script_path tcl2 1 12
textobject textdir Text01 1 5
audio audiodir spacemusic.au 1 5
unit_length 10
unit
timeline 0
##### Timeline 1 #####
name Chapter 0
choice .msg 0 dialog=dproc3 1 5
script $script_path tcl3 1 5
textobject textdir Text03 1 10
audio audiodir bubble1.au 7 9
unit_length 15
unit
name Unit1
choice .msg 0 dialog=dproc1 1 4
script $script_path tcl1 1 5
textobject textdir Text04 1 10
audio audiodir ah.au 7 15
unit_length 20
unit
timeline 1
#####
```

Figure 3.7 : Example of unit-definition file

3.5 Features of ITM

The following features of ITM make it useful for authors to create and present interactive multimedia documents:

1. Understandability - ITM is based on the traditional timeline model and it is easy for authors to understand it and structure interactive multimedia documents in a clear and concise manner. The logical structure of the document makes navigation easy and it is simple to trace back the path taken by the user. The temporal structure allows an author to completely specify all temporal relationships between objects. It allows the representation of both synchronous and asynchronous events in a unit. The documents created are easily interpreted and understood.

2. Modularity and abstraction - The interfaces provided by ITM are modular. The model provides independent interfaces for composition, presentation, interaction and control. Modularity helps in easy maintenance and reuse of components. The logical and temporal structures provide several layers of abstraction. Since the design of ITM is modular, it is easy to verify its features.

3. Maintainability - The interfaces provided by ITM are created as independent modules and are maintainable. Modifications can be easily made to the playback tool and the features of user interface. By means of the registration mechanism, the author can add new features to the model.

4. Reusability - Once the data objects are created, they can be stored in the database and reused to compose several multimedia scenarios. Similarly, the presentation structure of objects can be reused. Several data objects can have the same spatial component of presentation structure. Objects can have the same temporal structure, but different presentation structures.

5. User Friendly - The user interface for the playback tool allows the author to present the document, the user to view it and interact with the presentation, and allows the tool to control the interaction. It provides correct responses to user actions. It also starts the default response if the user does not respond within a specified time. Interactivity is supported at the lowest level by means of the registration mechanism. The author can register a procedure related to an event in a choice object. By means of the numeric code associated with the procedure, the

playback tool can monitor events generated by the user and provide the appropriate response by executing the registered procedure.

6. Configurability - All configuration variables are put into a module, so that the author can configure the model according to the hardware or software available. The author can also change the path to the various media objects.

7. Portability - Currently documents can be presented in X-windows environment on the Unix operating system. Since ITM is implemented using Tcl/Tk, it can be easily ported to any platform that has a port of Tcl/Tk available. Currently Tcl ports are available on many Unix platforms, MacOS, Windows '95, and Windows NT operating system.

The features provided by the document structure supported by ITM and the implementation details for the feature are listed in Table 3.3

Table 3.3 Features and Implementation details for ITM

ITM Layer	Feature	Implementation Details
Storage	Implicit Choice	Buttons on User Interface - Back, Forward, Pause, Home, Index, Help, Exit.
	Explicit Choice	Dialog box or message box informs user to generate an event. Event is bound to a response procedure. Multiple responses can be bound to an event.
	Active Text	Embedded Tcl code in a text file.
	Registration	Playback tool keeps track of the numeric code associated with each registered procedure.
	Script Object	Tcl code file
Composition	Unit	A list of all media objects with their temporal structure specified.
Presentation	Object Duration	Derived from Start and End Time
	Duration relationship	Derived from Start and End Time
	Interval Relationship	Derived from Start and End Time
	Synchronization mechanism	After time equal to length of unit, presentation of all media objects stops.
Run-time	Default-timeout action	Sequencing of units and timelines via Inter-unit and Inter-timeline temporal structure.
	Default-timeout on/off	Pause/resume button.

3.6 Limitations of ITM

The primary limitations of ITM are :

1. ITM does not support non-persistent data. Non-persistent data is defined as the data whose duration is not known until run time. Due to a fixed unit length in

ITM, the authors cannot directly link to a remote database and use media objects whose duration is not known in advance. To overcome this limitation, ITM must be modified. Unit length must be made variable, and the availability of non-persistent data must be monitored to use it as a media object in the unit.

2. ITM does not address the problems of distributed multimedia systems - eg. fault tolerance, network problems, etc.

The limitations of implementaion of ITM are :

1. Currently ITM runs only on UNIX platforms under X-windows. However, it is portable to any pltfom that has ports to Tcl/Tk.

2. ITM does not include video as a media object. However, video and other media types can be added to ITM. This would require changing the interfaces provided by the storage, composition, presentation, and run-time layers of ITM, but the changes are fairly simple.

CHAPTER 4

4. EVALUATION OF ITM

4.1 An Overview

The interactive timeline model is a software product, and both the author and the viewer are the users of this product. The author would use the model to create an interactive multimedia document, and the viewer would use the playback tool provided by the model to view the document and interact with it. The model has been designed by placing the author and the viewer at the center of the design process. We need to evaluate the model from the users' point of view to determine the usability of the model. This chapter explains the objectives of evaluation of ITM, describes how authors and viewers can evaluate the model, and presents evaluation results. Section 4.2 gives a background on usability testing. The objectives of evaluation of ITM, the approach to testing, and the testing methodology is described in section 4.3. Evaluation results are presented in section 4.4.

4.2 Background on Usability Testing

4.2.1 An Overview

The goals of ITM, the features it provides, and the environment in which it operates, are all derived from the authors and the viewers' viewpoint. To determine the usability of the model, and to evaluate the design decisions of ITM, we need to evaluate it from both the authors' and the viewers' point of view. In this section, we first define the term usability, and then state the goals of usability testing.

4.2.2 Usability

According to [13], the operational definition of usability includes one or more of the following four factors :

1. Usefulness - Usefulness concerns the degree to which a product enables a user to achieve his or her goals and is an assessment of the users' motivation for using the product at all. If a system is easy to use, easy to learn, and even satisfying to use, but does not achieve the specific goals of a specific user, it will not be used.

2. Effectiveness (ease of use) - Effectiveness is defined quantitatively, either by speed of performance, or error rate, and is tied to some percentage of total users. An example of such as measure would be "95 percent of all users will be able to load the software correctly on the first attempt in less than 10 minutes."

3. Learnability - Learnability has to do with the users' ability to operate the system to some defined level of competence after some predetermined amount and period of training. It can also refer to the ability of infrequent users to relearn the system after periods of inactivity.

4. Attitude (likability) - Attitude refers to the user's perceptions, feelings and opinions of the product, usually captured through both written and oral interrogation. Users are more likely to perform well on a product that meets their needs and provides satisfaction than one that does not. Typically, users are asked to rate and rank products that they test, and this can often reveal causes and reasons for problems that occur.

4.2.3 Usability Testing

Usability goals and objectives are typically defined in measurable terms of one or more of the four attributes described above. Usability testing employs

techniques to collect empirical data, while observing representative end users using the product to perform representative tasks. [13] emphasizes an informal approach to usability testing, which employs an iterative cycle of tests intended to expose usability deficiencies and gradually shape or mold the product in question. This type of testing is divided into four types of tests - exploratory, assessment, validation, and comparison tests, and the first three of these tests are associated with a particular phase in the product's life cycle, at which they are most effectively conducted. A brief description of when the tests are conducted, and objective of the tests follows.

1. Exploratory Test - The exploratory test is conducted quite early in the development cycle, when the specifications phase is complete and the design phase is just about to begin. The objective of this test is to examine or explore the effectiveness of preliminary design concepts, and to verify assumptions about the users' expectations of the product. This test is important because it is conducted at a point when critical design decisions set the stage for all that will follow. If the product begins with the wrong assumptions and faulty premises about the user, the product is almost guaranteed to have usability problems later.

2. Assessment Test - This test is usually conducted after the fundamental or high level design or organization of the product has been established. This test seeks to examine and evaluate how well the concept of the product has been implemented. Rather than just exploring the intuitiveness of the product, this test is interested in seeing how well a user can perform realistic tasks and in identifying specific usability deficiencies that are present.

3. Validation Test - This test, also referred to as the verification test, is intended to certify the product's usability. This test typically takes place late in the development cycle, much closer to the release of the product. This test evaluates how the product compares to some predetermined usability standard. These standards originate from usability objectives stated early in the project. Usability objectives are typically stated in terms of performance criteria and preference criteria. Performance criteria, such as speed and accuracy, determine how well and how fast can a user perform various tasks and operations. Preference criteria can be stated in terms of achieving a particular ranking or rating from the users. A major objective of the validation test is to evaluate, sometimes for the first time, how all components of a product work together. Another objective is to ensure that it does not have any major flaws before it is released.

4. Comparison Test - This test can be used in conjunction with any of the above three tests at any stage in the development life cycle. It is used to compare two or more alternative designs, such as different interface styles, or the current design of the manual with the proposed new design, or to compare your product with the competitor's. This test is used to establish which design is easier to use or learn and to better understand the advantages and disadvantages of different designs.

4.3 ITM Evaluation - Objectives and Methodology

4.3.1 Evaluation Objectives

An informal approach to testing was adopted to evaluate the design decisions of ITM, identify usability deficiencies existing in the ITM model, and to ensure that the model is easy to learn and use, is satisfying to use, and provides the utility and functionality that is valued by users of the model. A small number of users was used for evaluation to get a rough idea about the usability of the model and the playback tool. The model was evaluated informally in two phases of its development. Exploratory and comparison tests were carried out before and during the design phase of the model. Assessment and validation tests were

carried out after the design of the model. Exploratory tests were carried out after the specification phase and during the design phase of ITM. The objective of this test was to examine the effectiveness of preliminary design concepts of ITM, and to verify assumptions about the authors' and viewers' expectations of the model. Comparison tests were conducted throughout the design phase of the model. We focused on the assessment and validation tests to evaluate the design and usability of the model, and to validate the implementation of the model. We stated the performance and preference criteria to verify the usability of the model. The specific goals and methodology of assessment and validation tests are described in section 4.3.3.

4.3.2 Evaluation Methodology

To evaluate the model, we followed these basic elements of usability testing [13] :

1. Develop test objectives and methodology for assessment and validation tests.
2. Use a representative sample of end users (both authors and viewers).
3. Represent the actual work environment.
4. Observe end users who are using or viewing a representation of the model.

5. Collect quantitative and qualitative performance and preference measures.
6. Recommend improvements to the design of ITM.

We created a test plan for assessment and validation tests. The test plan included test objectives, criteria for evaluation, data to be collected (questionnaire for users), task list that we created for both authors and viewers, the role of the test participants, and the role of the test monitor (the person conducting the test). We conducted tests using a small sample of representative end users (2 authors and 3 viewers) to get feedback on the design of ITM and the usefulness of the playback tool. The users were faculty members and students of Electrical Engineering Department at Portland State University. Hence, the users' background and abilities were representative of the abilities of the model's end users.

4.3.2.1 Exploratory Test

The process for exploratory tests was quite informal. After the specification phase of the model, a preliminary version of the model's interface was developed for evaluation. Using this prototype we could explore the graphical user interface of the playback tool, present a simple document, comment on the presentation, interactivity etc. We explored the product both as an author, and as a

viewer. As users we suggested ideas on how to improve confusing areas. We attempted to answer typical user-oriented questions like -

1. What do authors think and conceive about using the model ?
2. Does the model's basic functionality have value to the author ?
3. Are the operations and navigation of the user interface intuitive ?
4. What type of prerequisite information does a user need to use the model ?
5. What functions of the model are easy to use, and which will require either help or written documentation ?

4.3.2.2 Comparison Test

Comparison tests were conducted informally to compare two or more alternative designs of the features of the model. The aim was to determine which aspects of the design of ITM are favorable and to establish which design is easier to use or learn. We also compared different interface styles for the user interface and compared the ITM model with the other models for interactive multimedia documents.

4.3.2.3 Assessment Test

The main objectives of this test were:

1. To evaluate how well the concept of the model has been implemented.
2. Use the data to identify specific usability deficiencies in the model.

The emphasis was on measuring how well the user is able to perform by collecting qualitative data. Rather than just exploring the intuitiveness of a product, we were interested in seeing how well the user could actually perform tasks. We asked some authors and viewers to perform tasks rather than simply walking through and commenting on the design of the model. We created realistic tasks for the authors, such as creating interactive tutorials and asked viewers to use the playback tool to present these tutorials, navigate through the tutorials using the playback tool or by using choice objects, get help from the user manual or on-line help etc. We made sure that all users got the same directions to do the same tasks. We observed the users perform their tasks and noted what problems they had. We did not interact with the users while they were performing these tasks, since there was more emphasis on the actual behavior of the user. The method of collecting qualitative data from users was informal. We asked the users to answer a questionnaire with the following questions :

1. How usable is the model - is it useful, easy to learn and use ?
2. How effective is the model in facilitating authors to create and present interactive documents ?
3. What is the opinion of the authors about the functionality of the model - does it meet their needs and expectations ? What features of the model were most useful to the author ?
4. How much did the viewer learn from a presentation ? How interactive is the playback tool ? Are there difficulties in navigating through the document ?
5. How did the viewers like the look and design of the playback tool ?
6. Is the documentation being utilized as designed ? Is it accessible ? Is all terminology clear ? Are there areas that require more explanation ? Are users more likely to access help or written documentation when confronted with difficulties ?
7. Is the help utilized as designed ? Are there difficulties in navigating, entry or exit ?

4.3.2.4 Validation Test

The main objectives of this test were :

1. To evaluate, how all components of the model work together, for example, how documentation, help, and software/hardware are integrated with each other. Since

the components were developed in isolation from each other, it was important to verify that they all work well together, and the model performs in the way it is expected.

2. To certify the model's usability by collecting performance and preference data and interpreting it.
3. To ensure that the model does not have any major flaws, and that all 'known bugs' have been 'fixed'.

The last objective was met by running tests on the model without the help of users. For first two objectives, tests were conducted informally a manner similar to the assessment test with two major exceptions :

1. Authors were given tasks to perform with no interaction or help from us. We observed if the model performed in the way it is expected, and if there is a seamless connection of software, help, and documentation. Can the authors and users move easily between the three elements ?
2. We collected quantitative data - both performance and preference data during and after the tests. Prior to the tests, performance and preference criteria were stated. For example, for the criteria for the performance data - "Time to complete a task" was that all authors must complete a given task in a given time, say 30

minutes. The criteria set up by us were subjective based on the performance of an expert user. For example, we established that if authors are able to complete a particular task in less than 30 minutes, the model is easy to learn and use, or if viewers spend no more than 5 minutes on looking for help for a particular topic in *on-line help* or the manual, the help is being utilized as designed , or if users completed more than 70% of the tasks correctly without assistance, the model is usable. A sample of some measures collected during a test is given below:

Performance data :

Time to complete each task.

Number and percentage of tasks completed correctly with and without assistance.

Number and percentage of tasks completed incorrectly.

Time required to access information in the manual.

Time required to access information in the on-line help.

Count of visits to the index.

Counts of 'negative comments'.

Preference data : Ratings and rationale concerning -

Usefulness of the model.

How well the model matched expectations.

Appropriateness of the model functions to users' tasks.

Ease of use overall.

Ease of learning overall.

Ease of accessibility.

Usefulness of the user interface, manual, help etc.

Examples of some tests conducted to collect the data are given below :

1. How much time do authors take to complete the same task, given the same directions to perform a task, such as authoring an interactive multimedia tutorial using the model.
2. How much time do different viewers take to view the presentation of the same tutorial.
3. On a scale of 1 to 10, how do different viewers rate the interactivity of the tutorial.
4. How many times does a viewer use choice objects in a tutorial to change the course of presentation of the tutorial ?
5. At what point, and how many times does a user refer to the user manual or on-line help during the presentation ?

4.4 Evaluation Results

Exploratory and comparison tests done during the specification and design phase of ITM helped us evaluate the preliminary design concepts, verified our assumptions about the users' expectations, and helped us refine the design of the model, and the user interface for the playback tool.

During assessment and validation tests, we collected performance and preference data from the users, which helped us assess the functionality of the model and its implementation and get an idea about the usability of the model. We observed that both the authors completed a task given to them in less than 30 minutes. The authors found the model easy to understand and use. Both authors and viewers referred to on-line help and the users manual and found it accessible and useful. All the users completed the tasks given to them without assistance from the test monitor. The viewers used the buttons on the user interface as well as the choice objects to navigate through an interactive tutorial. Different viewers take from 30 to 45 minutes to view the same tutorial, depending on how many times they navigate through the document.

Feedback from authors suggests that conceptually the model is easy to learn, but it would require more experience on the authors' part to put the model

into practice. The authors liked the "timeout feature" in the model, ie., the fact that all media objects are presented only for the duration of a unit. Also, the division of a timeline into units makes it easy for them to create the logical structure of the document.

The viewers liked the look and design of the user interface for the playback tool. No problems were encountered in using on-line help. Some users preferred to use help first and then the users manual. Viewers had no difficulty in navigating though the document using the buttons on the user interface or the choice objects presented. They suggested some improvements in the spatial layout of the document, like placement of images and graphics within the user interface window so that it does not overlap the text presented in the text widget.

A suggestion from the viewers was that this model can be used for two different applications. Firstly, it can be used for creation of a 'dynamic presentation', where the document plays itself back without user intervention, but the user can interact with it if he or she chooses to. Secondly, it can be used for creation of learning materials, for example, a tutorial, where a user is allowed to proceed at his or her own pace.

CHAPTER 5

5. CONCLUSIONS

5.1 The Interactive Timeline Model

Multimedia documents contain multiple media, like text, graphics, images, and continuous media like audio and video, which may be presented simultaneously or in some related manner in time. An interactive multimedia document not only integrates and presents continuous media, but also allows the user to interact with the running presentation. The inclusion of continuous media in interactive multimedia documents imposes new requirements on document representation and storage.

We stated the requirements of the model for interactive multimedia documents and described document models that fulfill some of these requirements. The present models of interactive multimedia documents lack features that allow the user to interact with the running presentation. The model for an interactive multimedia document should support the creation of logical, temporal, presentation, and control structure of the document.

We designed, and implemented the interactive timeline model (ITM) for creation and presentation of interactive multimedia documents. We had the following goals throughout the design and implementation of the Interactive Timeline Model :

1. To design a model that can support the creation of multimedia document structure.
2. To design a model that is functionally correct, robust, understandable, efficient, reliable, maintainable, configurable, portable, user friendly, verifiable, and has reusable components.

Our aim was to create a model that provides the facilities and operations multimedia authors expect, as well as those necessary to permit and ease the process of creation, presentation, and control of an interactive multimedia document. Hence the design of the model included the design of the logical, temporal, presentation, and control structures of the document. To present the documents, we included a playback tool in our design. To allow the user to readily interact with the presentation, we included the design of a user interface for the playback tool. Our interface decisions were motivated by the aim to

develop a system whose use matches the goals, needs, and expectations of the intended users.

The interactive timeline model extends the traditional timeline model to present multimedia scenarios. It increases the power of the timeline by adding a new media type, choice, which lets users to interact with a presentation. By means of the logical, temporal, presentation, and control structure supported by ITM, authors can create interactive multimedia scenarios, and integrate them into a presentation. ITM is a layered model that allows the authors to manage and reuse multimedia data effectively. The storage, composition, presentation, and run-time layers emphasize the separation of multimedia data from its logical, temporal, and control structures. ITM fulfills all the requirements of an interactive multimedia document model.

By means of the playback tool, users and other applications can interact with the presentation of a document. The playback tool has a user friendly interface and provides users with on-line help. The interfaces provided by ITM are maintainable and configurable, and are divided into author-interfaces and support interfaces. The authors can use the author-interfaces to create, present, and control

the presentation of an interactive document. Also, users can configure some parameters in the model according to their needs. By means of the registration mechanism, authors can create and register new procedures with the playback tool.

ITM allows creation, storage, and presentation of multimedia data in a clear and concise manner, yet is powerful enough to describe multimedia documents to the extent required by the authors.

5.2 ITM Evaluation

We evaluated the design of the model and got an idea about the usability of the model and the playback tool by conducting tests informally. We conducted exploratory and comparison tests during the specification and design phase of the model. We conducted assessment and validation tests by asking authors to create tutorials and by asking viewers to view the presentation of these tutorials. The evaluation of the model and its implementation resulted in refining the design of the model and assessing the usability of the model. The informal usability tests set the stage for extensive formal testing of the model to verify its usability. The users suggested improvements in the presentation structure of the model, and gave

suggestions on how to improve the model and make it more user friendly and interactive.

Feedback from authors and viewers suggests that ITM is a useful and powerful model to create and present interactive multimedia documents and would find good use to create 'dynamic presentations' as well as learning materials. On the basis of our research on the role of multimedia in distance education, we can say that this model can be effectively used to create and present distance learning materials in the Multimedia Instructional System project.

5.3 Future Work

Feedback from users can be used to improve the document structure supported by the model and make it more user friendly. The following features can easily be added to the model to improve it :

1. Spatial component of presentation structure - The spatial structure of the document can allow image objects to appear at a specific location in the user interface for the playback tool. The authors can be allowed to configure some variables in the model that allow them to change the spatial structure provided by

the model. This modification can be easily done using the “place” command in Tcl.

2. ITM does not include video as a media object. However, video and other media types can be added to ITM. This would require changing the interfaces provided by the storage, composition, presentation, and run-time layers of ITM. The interfaces provided by ITM are modular, hence the inclusion of new media types is not difficult.

3. ITM has been tested only on UNIX platforms under X-windows. It can be tested on other platforms that have ports to Tcl/Tk available.

4. The evaluation of ITM was done informally because of limited resources. A small number of users was involved in evaluation in a short period of time. The evaluation should be done using more extensive, structured and formal testing methods. Formal usability criteria must be established and tight controls must be employed to evaluate the degree to which the model meets specific criteria. Formal testing requires more resources in terms of time, participants involved and detailed testing methodology, hence informal testing is usually done [13]. A hypothesis must be formulated for formal testing. For example, "Design A of a feature of the model will improve the speed of experienced users more than design B of the same feature". Test participants must be chosen by random

sampling of the target population. The sample of users must be of a sufficient size to measure statistically significant differences between groups. Comparison tests should be conducted to compare the model with commercial tools to establish which design is easier to use or learn. Prior to the formal validation test, benchmarks or standards for the tasks of the test should be identified or developed.

5. A limitation of ITM is that it does not support inclusion of media objects whose duration of presentation is not known in advance (non persistent data). To support this feature, the unit length must be made variable, and the availability of non-persistent data must be monitored to use it as a media object in the unit. This change requires adding interfaces to the model to check the availability of data at the end of a unit. If the data is not available, the length of the unit should be increased. This modification requires extensive changes to the interfaces provided by the presentation and run-time layers of ITM.

6. A new button on the user interface - "View Source", can allow users to view the unit-definition file for the document being presented. This would allow authors to present a document and simultaneously learn about the mistakes in the unit-definition file. Addition of this feature is easy since the elements of the user interface are configurable.

REFERENCES

- [1] Ahmad Karmouch, and James Emery, "A Playback Schedule Model for Multimedia documents", *IEEE Multimedia*, Spring 1996, pp. 50 - 61
- [2] Ahmad Karmouch, Nael Hirzalla and Ben Falchuk, "A Temporal model for Interactive Multimedia Scenarios", *IEEE Multimedia*, Spring 1995, pp.24-31
- [3] M.J. Perez-Luque and T.D.C. Little, A Temporal Reference Framework for Multimedia Synchronization", *IEEE Journal on Selected Areas in Communications*.
- [4] M.C. Buchanan and P.T. Zellweger, "Automatic Temporal Layout mechanisms", *Proc. ACM Multimedia* , August 1993, pp. 341-350.
- [5] T.D.C. Little and A. Ghafoor, "Synchronization and Storage Models for Multimedia Objects", *IEEE Journal on Selected Areas in Communications*, Vol.8, No.3, March 1990, pp. 413-427
- [6] Gehard A. Schloss, and Michael J. Wynblatt, "Providing definition and temporal structure for multimedia data", *Multimedia Systems*, March 1995, pp. 264-277
- [7] H. Khalfallah, and A. Karmouch , "An architecture and a data model for integrated multimedia documents and presentational applications", *Multimedia Systems*, March 1995, pp. 238-250
- [8] Lynda Hardman, Dick C. A. Bulterman, and Guido Van Rossum, "The Amsterdam Hypermedia model", *Communications of the ACM*, February 1994, Vol. 37, No. 2, pp. 50-62
- [9] Frank Halasz, and Mayer Schwartz, "The Dexter Hypertext Reference Model", *Communications of the ACM*, February 1994, Vol. 37, No. 2, pp. 30-39

- [10] R. Rossum et al., "CMIFed: A Presentation Environment for Portable Hypermedia Documents", *Proc. ACM Multimedia 93*, ACM Press, New York, 1993, pp. 183-188.
- [11] L. Weitzman and K. Wittenburg, "Automatic Presentation of Multimedia Documents Using Relational Grammars", *Proc. ACM Multimedia 94*, ACM Press, New York, 1994, pp. 443-451.
- [12] Carlo Ghezzi, Mehdi Jazayeri, and Dino Mandrioli, *Fundamentals of Software Engineering*, Prentice Hall Inc., 1991
- [13] Jeffrey Rubin, *Handbook of Usability Testing : How to Plan, Design, and Conduct Effective Tests*, John Wiley & Sons, 1994
- [14] R. C. Shank, "Active Learning Through Multimedia", *IEEE Multimedia*, Vol. 1, No. 1, Spring 1994, pp. 69-78.
- [15] Beverly Park Woolf, Wendy Hall, "Multimedia Pedagogues", *IEEE Computer*, May 1995, pp. 74- 80.
- [16] Murray Turroff, "Designing a virtual Classroom[TM]", <http://it.njit.edu/njIT/Department/CCCC/VC/Papers/Design.html>
- [17] Brent B. Welch, *Practical Programming in Tcl and Tk*, Prentice Hall 1995
- [18] Vibha M. Nayyar and Dr. Michael A. Driscoll, *ITM Users Manual*, Internal publication, Department of Electrical Engineering, Portland State University, 1996