studocu

Chapter-7

Multimedia System (Pokhara University)

Scan to open on Studocu

Studocu is not sponsored or endorsed by any college or university
Downloaded by staven william (williamstaven428@gmail.com)

Chapter 7 - Documents, Hypertext and MHEG

A document consists of a set of structural information that can be different forms of media, and during presentation can be generated or recorded.

Documents

A multimedia document is a document which is comprised of information coded in at least one continuous (time-dependent) medium and in one discrete (time independent) medium. The media are synchronized. A multimedia document is closely related to its environment of tools, data abstractions, basic concepts and document architecture.

Continuous and discrete data are processed differently: text is processed within an editor program as a type of a programming language; a motion picture can be manipulated with the same editor program only through library calls.

Basic *system* concepts for document processing use multimedia abstractions and also serve as concepts for the information architecture in a document.

Document Architecture

In order to exchange documents the content of the document as well as its structure needs to be communicated. This requires the use of standard format for the documents so that it can be communicated to wide range of users. The current formats on the process of standardization are *Standard Generalized Markup Language (SGML)* and the *Open Document Architecture (ODA).* It means a standard architecture of document is required. Document architecture describes the connections among the individual elements represented as models.

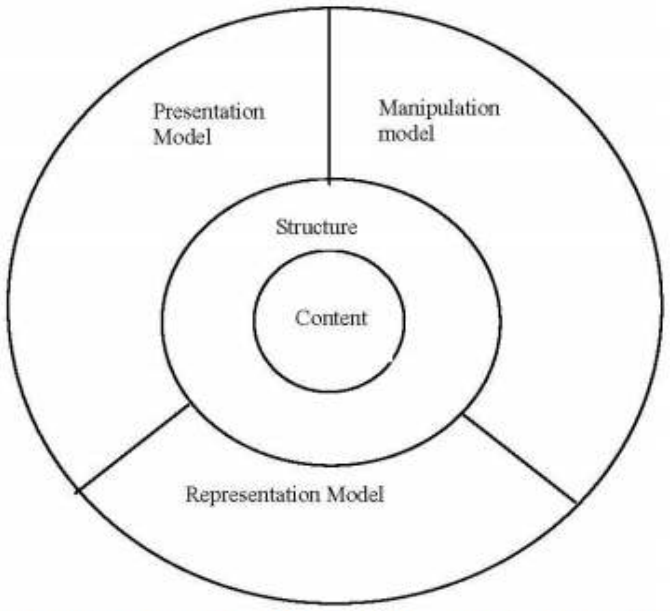


Figure: Document architecture and its elements

The *manipulation model* describes all the operations allowed for creation, change and deletion of multimedia information.

The *representation model* includes the relations between the individual information elements which need to be considered during presentation. It defines:

This document is available on studocu

Downloaded by staven william (williamstaven428@gmail.com)

- ✓ *protocols* for exchanging information among different computers
- ✓ *formats* for storing the data

The *presentation model* describes how the content of the document is displayed or presented before the users. The Structure is described by the links and the synchronization parameters of the document. The content is the actual information that the document carries.

Manipulation of Multimedia Data
The manipulation tools for the multimedia document or data are the editors, desktop publishing programs, text, image and audio processing programs.

The information included in the document classifies it into different document types like letters, articles, invitation etc. However the transformation from the actual information to its final representation behaves according to rules specific to the document architecture. While processing the document exists in a process able representation. The subsequent formatting process determines the layout of the document. The result is a final representation of the document.

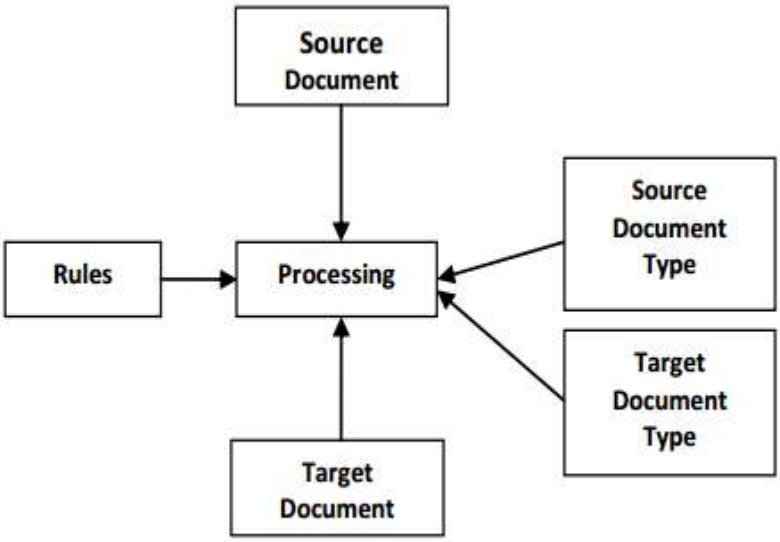


Figure: Processing of a document – from the information to the presentation

Hypertext, Hypermedia and Multimedia

Ordinary documents are transformed into a linear document and the structural information is integrated into the actual content. In the case of hypertext and hypermedia, a graphical structure is possible in a document which may simplify the writing and reading processes.

The reading of the document may not always be sequential. For e.g. when a reader is reading about how internet works, he might want to learn about the protocols which the sequential document does not contain in such case it is possible to provide the link to the key terms from the document to some other related documents. Similarly when reading a biography the user may want to view the pictures of the person, or the videos related to him. It may not be possible to include that additional information in the same document but it is still possible to provide a link to the video or the audio.

Downloaded by staven william (williamstaven428@gmail.com)

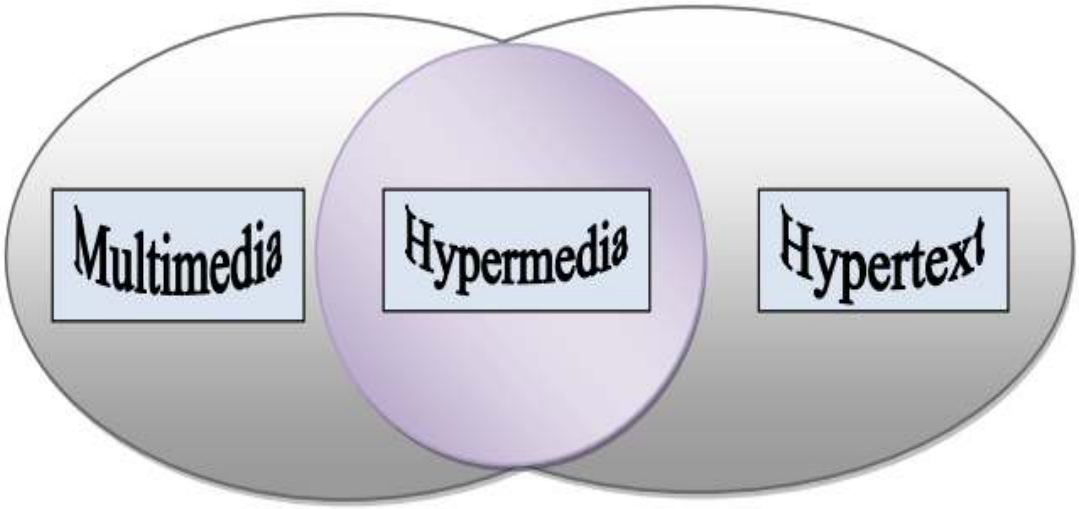


Figure: The hypertext, hypermedia and multimedia relationship

Hypertext can be a word, phrase, or a sentence that provides a non-linear link from a document to some other document. A *hypertext system* is mainly determined through non-linear links of information.

Hypermedia on the other hand is nonlinear way of presenting information that allows users to access related works or images from a single computer screen. For example, a user reading an encyclopedia entry on jazz could also hear excerpts from recordings, read biographies of jazz artists, and view photos of them. Apple Computer Inc.'s

Hyper-card is the best-known example of hypermedia. Presumably, this type of interface is similar to normal human cognitive processes. It is in fact the hypertext extended so that the link could be provided with the help of image or other data structure different than the text.
A *hypermedia system* includes the non-linear information links of hypertext systems and the continuous and discrete media of multimedia systems.

A *multimedia system* consists of information which is coded at least in a continuous and discrete medium.

Non-Linear Information Chain
Hypertext and hypermedia have a major property a *non-linear information link.* There exists not only a reading sequence, but also the reader decides on his/her reading path. The reader can start from a page with a notion hypertext, then go through a cross reference to systems and finish with audio-visual information.

A hypertext structure is a graph, consisting of nodes and edges.

- ✓ The *nodes* are the actual information units. They are for example the text elements, individual graphics, audio or video LDUs. The information units are shown at the user interface mostly in their own windows.
- ✓ The *edges* provide links to other information units. They are usually called pointers or links. A pointer is mostly a directed edge and includes its own information too.

This document is available on studocu
Downloaded by staven william (williamstaven428@gmail.com)

Anchor

Exploring a non-linear document is *navigation.* The document is linked with other related document with the help of pointers. This origin of a pointer is called an anchor. Representation of the anchor can be

- ✓ A *media-independent* representation can happen through the selection of general graphical elements, such as buttons.
- ✓ In a *text,* individual words, paragraphs or sections of text can be used as pointer. These texts are visually different from rest of the document. The user is flown to the other document with a click to these texts.
- ✓ In *images,* specific graphical objects or simply areas are defined as selection objects. A specific marking can occur through a color or stripe.
- ✓ In a *motion video,* media-independent representations of the anchor are preferred. A timely selection is supported.
- ✓ With respect to *audio,* a media-independent solution is used for e.g. by using descriptive text or an image of the size of an icon.

Application Areas of Hypertext, Hypermedia and Multimedia

- ✓ While giving lecture on hypermedia documents.
- ✓ In some classical computer applications like the "help" function
- ✓ In the area of commercial applications, repair and operational manuals can be found.
- ✓ The organization of ideas, brainstorming and the generation of scientific documents count, for example, as intellectual applications.
- ✓ Tourist information systems and interactive science-fiction movies count on the areas of entertainment and free-time activities.

Systems: Architecture, Nodes and Pointers

Architecture

Presentation Layer

At the upper layer, the presentation layer, all functions connected to the user interface are embedded. Here, nodes and pointers are mapped to the user interface.

Hypertext Abstract Machine

HAM is placed between the presentation and storage layers. It knows the structure of the document and it expects database related functions from the lower layers.

Storage Layer

It is the lowest layer. All functions connected with the storage of data, i.e., secondary storage management, belong to this layer.

Nodes

A node is an information unit (LDU) in a hypertext document.

- ✓ The maximal stored *data amount* can be *limited* and mapped onto the screen size.
- ✓ Window-based systems with an *unlimited data amount* per node are the alternative. Forward and backward scrolling of pages is offered analogous to other windows at the user interface.

Pointers

Pointers are the edges of a hypertext graph. Pointers can be

- ✓ *Simple pointers* link two nodes of the graph without containing any further information.
- ✓ *Typed pointers* contain further information in addition to the link. Usually a tool tip text or a label is used with typed pointers.
- ✓ *Implicit Pointers* determine the relation between nodes automatically.
- ✓ *Explicit Pointers* is created by the author himself

Downloaded by staven william (williamstaven428@gmail.com)

When is the destination of a pointer specified?

- ✓ In the classical case, the pointer is created during the generation of hypertext document, and hereby the origin of the destination node is determined. The author determines explicitly the links of the information units during document processing.
- ✓ A destination node can be determined first by using the pointer, i.e., during reading. The author specifies an algorithm for the creation of the pointers, but they are determined first during the reading depending on the context.

Tools

The tools can be *editors* for processing the information. *Search tools* allow the search of desired information. *Browser* allows a shortened but clear representation of the nodes and edges. The nodes are described media-dependently. *Backtracking* prevents the users from getting lost in the hyperspace.

Document Architecture SGML

Standard Generalized Markup Language (SGML) is a standard for the uniformity in the content and their representation in the document. The Content of the document is described within the tags. SGML determines the form of tags, but it does not specify their location or meaning. It is basically a set of rules that break a document into parts and identify the different parts of the document. It basically defines the syntax or the structure of the document's content. However SGML does not provide semantics.

Example:

<BOOK>
<TITLE> Multimedia Communication </TITLE>
<AUTHOR> Jerry D. Gibson </AUTHOR>
<PRICE> NRs. 1000 </PRICE>
<PUBLICATION> PHI </PUBLICATION>
</BOOK>

Processing of SGML document

Processing of SGML document is divided into two processes. Only the formatter knows the meaning of the tag and it transforms the document into a formatted document. The parser uses the tags, occurring

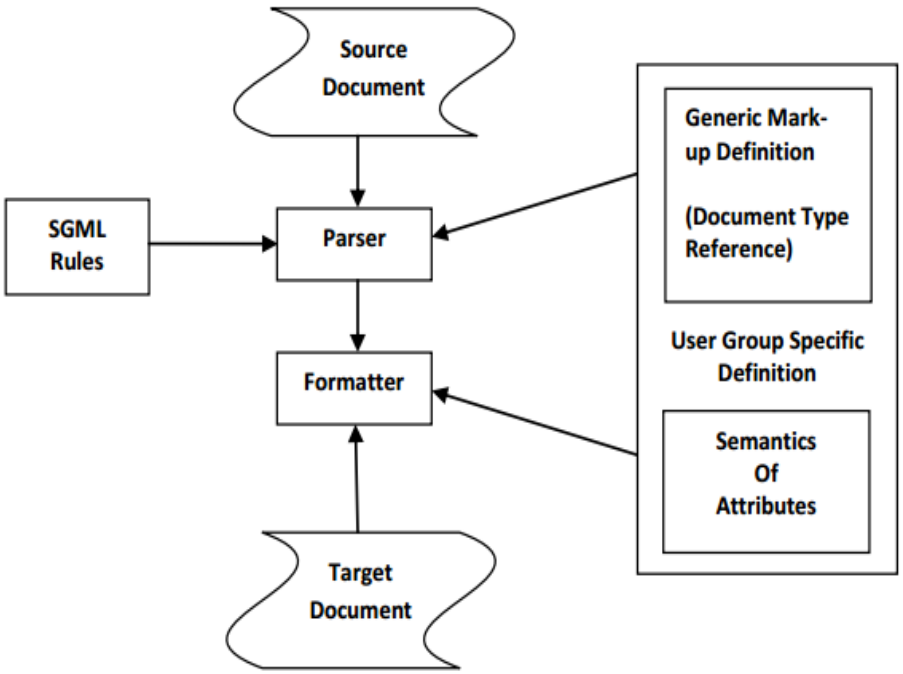


Figure: SGML Document processing - from the information to the presentation

This document is available on studocu
Downloaded by staven william (williamstaven428@gmail.com)

in the document, in combination with the corresponding document type.

Tags are divided into different categories

- ✓ The *descriptive markup (tags)* describes the actual structure always in the form:
  <start-tag> information </end-tag>
- ✓ The *entity reference* provides connection to another element. This element replaces the entity reference. The following example shows entity reference in a mathematical context:
  x^2
  &square x … should be x
- ✓ The *markup declarations* define the elements to which an entity reference refers. Squaring a variable x, square is defined as
  <!ELEMENT square (…)>
- ✓ Instructions for other programs in a text are entered through *processing instructions.* Using processing instructions, different media can be inserted.

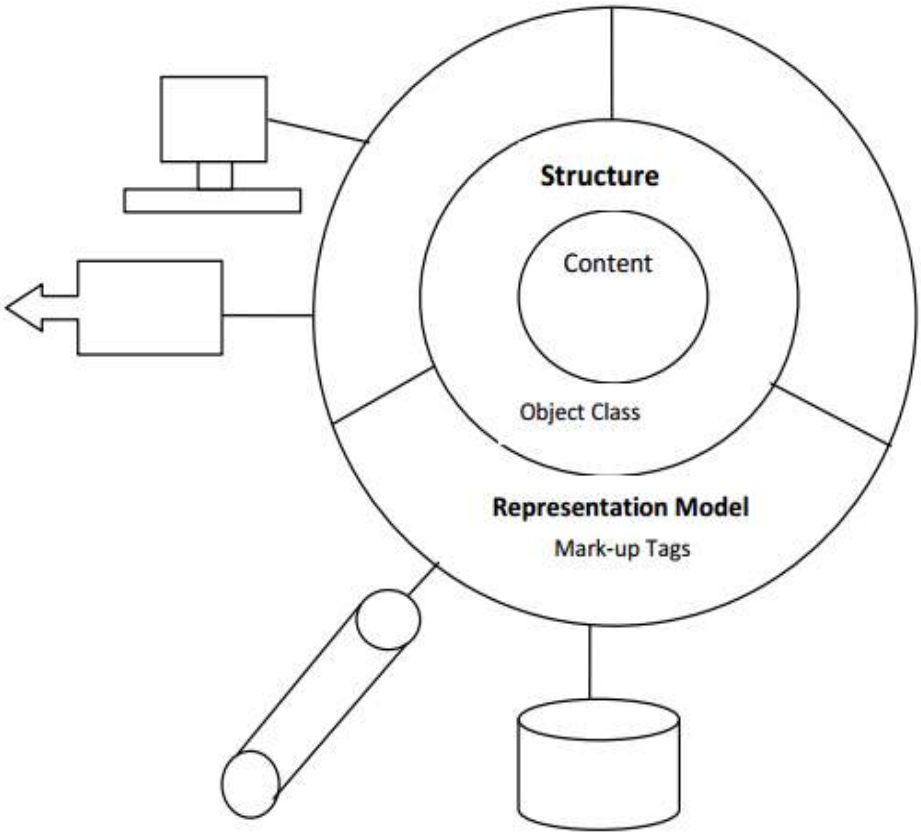


Figure: SGML: Document architecture - emphasis on the representation model

Document Architecture ODA

The *Open Document Architecture (ODA)* is a standard document file format created by the ITU-T to replace all proprietary document file formats. ODA defines a compound document format that can contain raw text, raster images and vector graphics. The documents have both logical and layout structures. Logically the text can be partitioned into chapters, footnotes and other sub elements, and the layout fills a function similar to Cascading Style Sheets in the web world.

Downloaded by staven william (williamstaven428@gmail.com)

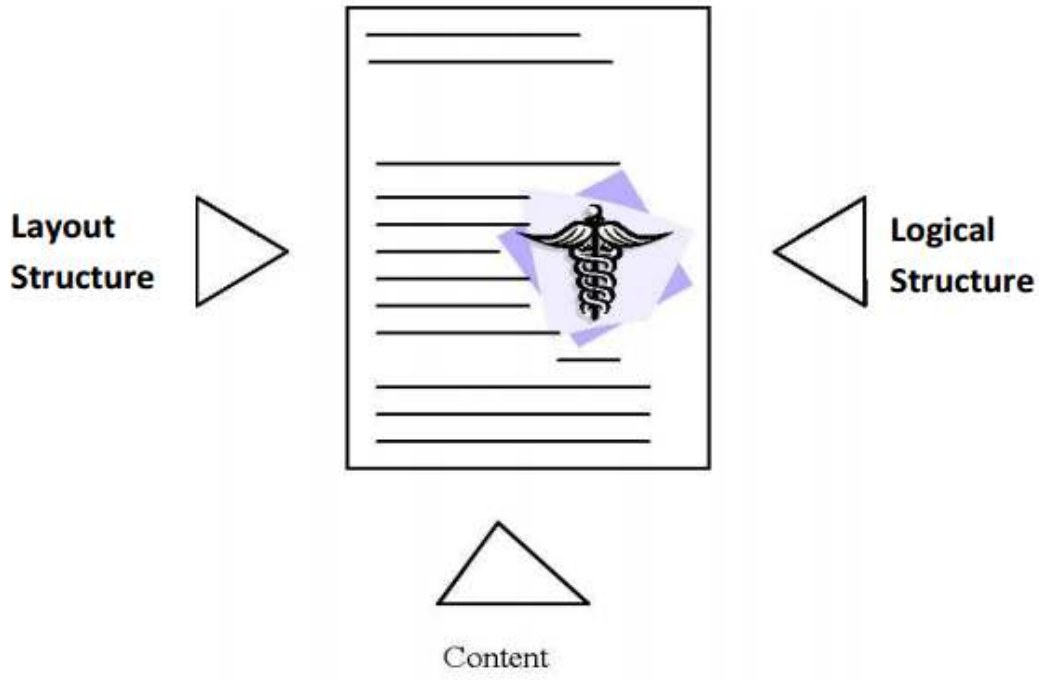


Figure: ODA: Content, layout and logical view

Figure: ODA: Content, layout and logical view

The binary transport format for an ODA-conformant file is called Open Document Interchange Format and is based on abstract syntax notation one (ASN.1). Thus the main property of ODA is the distinction among content, logical structure and layout structure. This is in contrast to SGML where only a logical structure and the contents are defined. ODA also defines semantics. The following figure shows the layout, logical and content aspect of the linked documents. Each of these views represent one aspect, together we get the actual document.

Content Portions

The content of the document consists of Content Portions. These can be manipulated according to the corresponding medium. The *content architecture* describes for each medium:

- ✓ the specification of the elements,
- ✓ the possible access functions and
- ✓ the data coding

Individual elements are the Logical Data Units (LDUs), which are determined for each medium.

Layout Structure and Logical Structure

The *layout* structure specifies mainly the representation of a document. It is related to a two dimensional representation with respect to a screen or paper.

The *logical* structure includes the partitioning of the content. Here, paragraphs and individual headings are specified according to the tree structure. The presentation itself is built around the tree model.

This document is available on studocu

Downloaded by staven william (williamstaven428@gmail.com)

Consider the following example:
Document = preamble body postamble
Header = document name
Footer = Page Number
Body = chapter1, chapter 2
Chapter1 = title, paragraph, picture, table, picture
Chapter2 = title, picture, table, text

In the example given above the document has the preamble and postamble, the header has the name of the document, and footer of each document has the page number, the body consists of the content as chapter1, chapter2 etc. Each chapter1 has the title of the content, paragraph, picture, and tabular data and picture again and so is with chapter2.

ODA distinguishes the following the layout and logical structures:

- ✓ The generic logical and generic layout structures include a set of default values. For example, a paragraph can be specified with *LeftHandOffset = 0.*
- ✓ The *specific* logical and *specific* layout structure describes a concrete document. They are linked to the generic structure. For example, a concrete paragraph can be defined with the *LeftHandOffset =1 cm.*

An ODA document can be of three types:

1. A *formatted document* includes the specific layout structure, and eventually the generic layout structure. It can be printed directly or displayed, but it cannot be changed.
2. A *processable document* consists of the specific logical structure, eventually the generic logical structure, and later of the generic layout structure. The document cannot be printed directly or displayed. Change of content is possible.
3. A *formatted processable document* is a mixed form. It can be printed, displayed and the content can be changed.

ODA and Multimedia

Multimedia requires, besides spatial representational dimensions, the *time* as the main part of a document. For the inclusion of continuous change is the standard is necessary which basically calls for an extension. The following changes will occur:

Contents
The *content portions* will change to *timed content portions.* The duration does not have to be specified earlier. These types of content portions are called *Open Timed Content Portions.* In case of a *Closed Timed Content Portion,* the duration is fixed. E.g. song

Structure
Operations between objects must be extended with a time dimension where the time relation is specified in the root node r in proportion to the child nodes c1, c2.

Content Architecture
Many functions are very often device-dependent. One of the most important aspects is a compatibility provision among different systems implementing ODA.

Logical Structures
Extensions for multimedia of the logical structure also need to be considered. For example, a film can include a logical structure.

Downloaded by staven william (williamstaven428@gmail.com)

Layout Structure
The layout structure needs extensions for multimedia. The time relation by a picture and audio must be included. Further, question such as *when will something be played?, From which point?,* And *With which attributes and dependencies?* must be answered.

MHEG (Multimedia and Hypermedia Information Coding Expert Group)

Several cross platform video and audio standards have been established including still and motion JPEG, and a number of different MPEG standards. So far, there has been no standard method of bringing all these formats together to produce multimedia presentations. The MHEG model aims to solve this by providing a system independent presentation standard for hardware and software engineers and presentation authors to conform to. In this way, a presentation created on one hardware platform should be viewable on others.

What is MHEG?
MHEG is an abbreviation for the Multimedia and Hypermedia Experts Group. This is another group of specialists, eminent in their field which has been set up by ISO, the International Standards Organization. This group was set the task of creating a standard method of storage, exchange and display of multimedia presentations.

Its basic goals are:

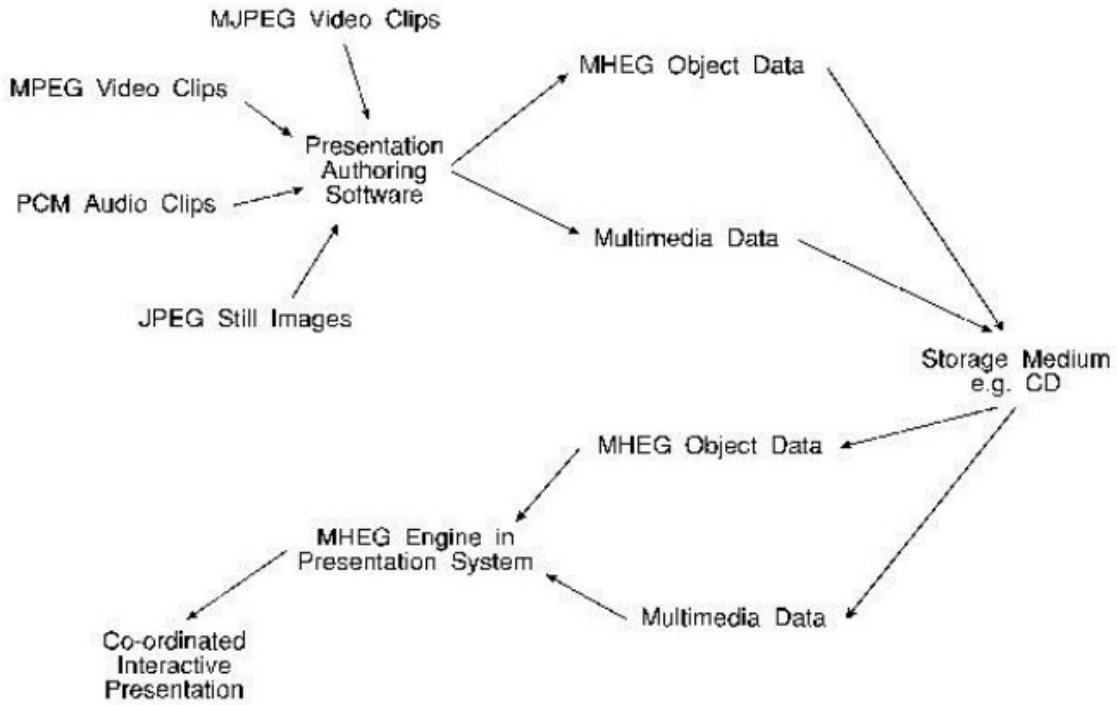
- ✓ To provide a simple but useful, easy to implement framework for multimedia applications using the minimum system resources.
- ✓ To define a digital final form for presentations, this may be used for exchange of the presentations between different machines no matter what structure or platform.
- ✓ To provide extensibility i.e. the system should be expandable and customizable with additional application specific code, though this may make the presentation platform dependent

The Reasoning behind MHEG
In order for products to have maximum commercial success, they must appeal to the largest number of consumers possible. There are various multimedia presentation packages available, but they are proprietary and do not work across different hardware platforms. This means that an author, who wants to publish an interactive multimedia book for example, will have to produce several versions that comply with different standards, or risk losing potential customers. The MHEG standard would do away with all this, allowing an author to produce his/her work in one universally acceptable format. It also has advantages for the hardware and software suppliers. They are able to concentrate on producing the one standard MHEG engine rather than an engine for every presentation standard that is available.
Applications of MHEG
The applications to which MHEG may be put are growing all the time, as people dream up more and more applications of multimedia. Here are a few examples:

- ✓ CD-ROM based encyclopedias
- ✓ Interactive books for learning
- ✓ Video and news on demand systems
- ✓ Interactive home shopping

Structure of MHEG
MHEG defines the abstract syntax through which presentations can be structured. This is the definition of data structures and the fields in those data structures, through which two computers may communicate. In the case of a multimedia book, these two computers would be that of the author and that of the user.

This document is available on studocu
Downloaded by staven william (williamstaven428@gmail.com)



The MHEG model is object orientated, and defines a number of classes from which object instances are created when a presentation is designed. There are several classes, and these are used to describe the way video is displayed, audio is reproduced and how the user can interact with the ongoing presentation. The relationship that is created between instances of these classes' forms the structure of the presentation. In addition to just replaying existing multimedia data MHEG also defines some types of its own. e.g. an MHEG compliant system is able to overlay titles onto video scenes. It is also able to display menus and buttons to allow the user to make choices.

There are several different types of class defined by MHEG. Some are concerned with the structure of the presentation and grouping of objects, whilst others are involved in the interchange of information between machines.

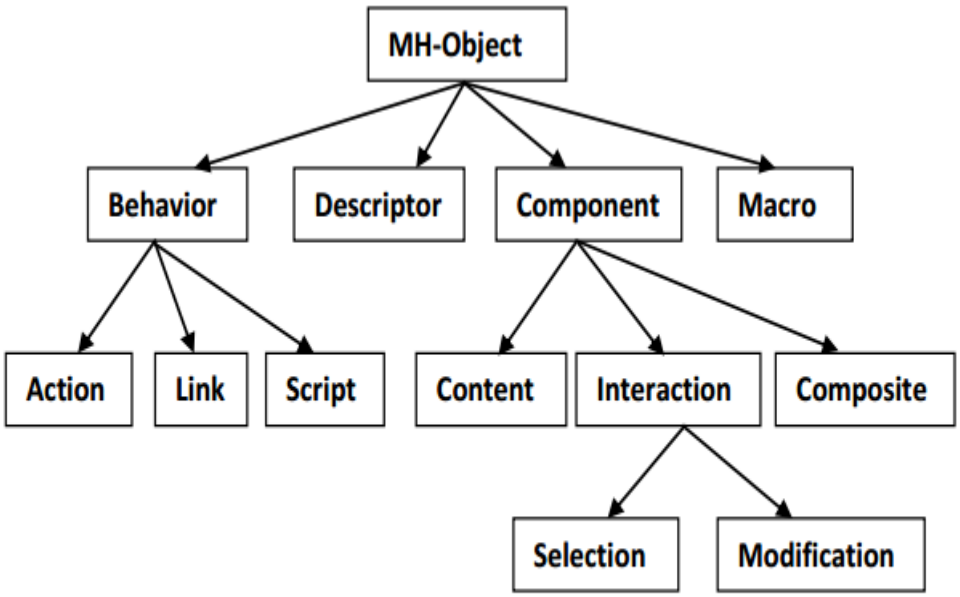


Figure: Class hierarchy of MHEG objects

Downloaded by staven william (williamstaven428@gmail.com)

Content Class:
Content classes are each piece of multimedia data e.g. video or audio clip has its own MHEG object. If the amount of data involved is small (a textual title), then it may be contained in the MHEG object itself. If not, then the MHEG object will give a reference to the data e.g. a disk filename. Depending on the form of the data, extraction of smaller sections of the data may be possible e.g. one audio channel from an MPEG audio/video clip.

Behavior Class:
Classes under this heading control how and when data is presented to the user. They allow synchronizing of events and user interaction.

Action Class:
The action class allows events to be triggered sequentially or in parallel e.g. the replay of several clips of video one after the other.

Link Class:
The link class establishes relationships between events and objects i.e. what actions to take on what objects in response to a particular event. A link object consists of a set of links.

Script Class:
It determines the behavior of the objects. It supports MHEG presentation in run-time environments.

Selection Class:
The selection class provides the possibility to model an interaction as a selection of a value from a predefined value set.

Modification Class:
It serves as the input and manipulation of data. No value set is predefined by a modification object.

Composite Class:
It has the task of composing all the necessary objects from the previously described classes into a presentation. If a single object is included or referenced, each composite object behaves as a *container.*

References:

- ✓ Multimedia: Computing, Communications and Applications", Ralf Steinmetz and Klara Nahrstedt, Pearson Education Asia
- ✓ "Multimedia Communications, Applications, Networks, protocols ad Standards", Fred Halsall, Pearson Education Asia
- ✓ "Multimedia Systems", John F. Koegel Buford, Pearson Education Asia

Assignments:

(1)What do you mean by SGML? Why is it required?
(2)Explain Hypermedia system with an example.
(3)Draw and discuss the multimedia Document Architecture.
(4)What are Hypertexts and Hypermedia?
(5)Differentiate between SGML and ODA document architecture.
(6)What do you mean by ODA? Why is it required?

This document is available on studocu

Downloaded by staven william (williamstaven428@gmail.com)

Downloaded by staven william (williamstaven428@gmail.com)