Method Designing for Multimedia and Hypermedia Application

R. N. Jugele
Department of Computer Science,
Science College, Congress Nagar,
Nagpur. Maharashtra
rn_jugele@yahoo.com

Dr. V. N. Chavan
Head, Department of Computer Science,
S. K. Porwal College, Kamptee,
Dist : Nagpur. Maharashtra

Abstract

Method is a multifaceted activity. Multimedia information technology has become popular, conventional database system do not offer adequate support for method, modeling, indexing and manipulation of multimedia data. Here we present details of its implementation and discuss the conclusions.

Keywords

Multimedia, real-time, object, script, level, virtual room, MHEG, method, Multimedia information object, style, pairs.

1. Introduction

This paper describe the method for application development, it shows the design criteria, and describes how the development method is designed. The considerations in these paper can as well be considered indications of how applications should be developed, for this is inherent to developing specific applications based on a general method. Finally picturizes the application development method based on architectural application composition.

2. Application Development Method Criteria

The method which is designed must fulfill several criteria which cover either general considerations or specific architectural concerns.

2.1. Method Design

This method is not a magic formula for converting the functional and logical requirements into the one and only correct application. It is just a disciplined approach for a structured process to guide the developer of multimedia applications in dealing with integrated or distributed presenting, representing, processing and handling.

2.2. Overall Aim

This method is to be both practical and usable in a long term for a broad range of developers. So, it must be simple, accessible and open. According to [1], this openness can be:
(a) in time, signifying openedness (modifiability and extensibility).
(b) in space, signifying distributivity.
(c) in technology, signifying heterogeneity.

3. Design of an Application Development Method

Method is described with respect to the model[4].

3.1. Presentation

Presentation is a system issue and it is treated below.
One may think of three major actors: the application, the interface and the system. The interface couples the application and the system. In this method it is assumed that the application is a set of flexible and fixed functional modules, identified as virtual rooms (VIR). The VIRs...
consist of sets of cooperating operational pairs (OP) which involve the multimedia information objects (MIO)[5].

The operating system (OS) interacts with the application through the application control program (ACP)[6], which interfaces the system through the OS. The operating system cares for scheduling, switching, networking and (synthetic and continuous) interflow synchronization. It cares for openness in space (distributivity). The application control program is responsible for object management and (synthetic and continuous) intraflow synchronization. It cares for openness in technology (heterogeneity). If an analogy can be made, then the OS could somewhat be compared with MSDOS and the ACP with Windows.

In this method it is assumed that the system is a set of device descriptions and driver programs (DDDPs) together with general devices. These devices can be image servers (display, scanner, video effect DSP), sound servers (speakers, microphone, CD-player, audio effect DSP), database servers (hard disk, CD-ROM), print servers or networked remote servers (printing, mail). The system consists of hardware devices (like scanners or screens) and their DDDPs.

This view of the complete system issue is an extensive elaboration of the principles introduced in [2] which describes DDDP and OS issues. The split of the ACP from the OS as well as the operational pairs and the virtual rooms are added features as shown in figure 1.

![Figure 1. Architectural Component](image)

3.2. Representation

The Object-Oriented methodology fits requirements of active, autonomous and reusable information objects this standard is described according to these principles, thus encapsulating distracting details allowing for inheritance-inherent abstraction levels and homogeneous object description. The representation of MHEG information objects is done at four levels (Figure 2).

(a) an informal text description.

(b) an object-oriented definition of an object’s structure in agreed upon semantics.

(c) a base notation for the structure of the representation.

(d) a base coded representation according to predefined coding rules.

![Figure 2. Representation Methodology](image)

The alternative representations at each of these levels are equivalent and can be translated to each other through isomorphism beyond the scope of this standard. Thus, this standard does provide some structures and semantics to develop a common understanding but it does not intend to provide all semantics for using application based on standardized objects. Representations at level-c and level-d may be done using Abstract Syntax Notation 1 (ASN.1).

A concise informal description at level-a provides for an initial intuitive understanding of the standardized elements in the representation of an object. Then at level-b with the help of a more precise Object-Oriented definition, further insight is given in class decadency, hierarchy and behavior. Here too, an object’s semantics and structure are described according to the following pattern: attribute name, semantics, type and status (mandatory, optional and default values).

On an object, a using application may invoke those operations which are listed at this level. The formal structure at level-c gives an object both body and boundary. At the same time, an encoding is provided by this standard when the actions are invoked by MHEG link object. No encoding is provided for operations invoked by a using application. Base coded representation is done according to the Basic Encoding Rules of ASN.1; here, at level-d, the following techniques are used:

(a) represent each MHEG object class by a separate ASN.1 module, formally accessible by a unique object identifier (not encoded in the MHEG object).

(b) represent the inheritance of MHEG object classes attributes by import/export facilities.
Note that due to the Object-Oriented inheritance property, super classes export and subclasses import object attributes.

(c) identify each MHEG object instance uniquely by its classes’ object identifier and an individual interchange number[5].

3.3. Handling

Distributed communications handling is a core topic in the application development method.

a. Transparency
While designing an application, we have to consider keeping it transparent to media concurrency and information replication. At the same time the application should be open in space and technology towards distributives of and heterogeneity in hardware, software and data. This openness is constrained by several factors, like the fact that at a remote resource, the prime user may prohibit information retrieval (for reasons of security, privacy, or autonomy control).

b. Synchronization
In multimedia applications, the problem of synchronization must be dealt with. Synchronization primitives must be formulated which can be applied at specific synchronization points.

3.4. Processing

Data processing is a core topic in the application development method.

a. Encapsulation
An application should consist of general functionality modules which are able to encapsulate media specific properties. Since any specific multimedia application is somehow involved in communications it must be able fulfill:

(a) the production role by outputting multimedia information objects in a certain representation.
(b) the consumption role by inputting objects in a certain presentation.

4. Architectural Application Composition

An application should realize some functionality in one of the working environments. In this view, an application is a set of virtual rooms which group cooperating operational pairs (environment; function) that define some scriptware pairs are applied at MHEG information objects.

With reference to the technical environment model, applications come standard:

(a) with selected multimedia information objects (Object Level).
(b) with developed operational pairs (Script Level).
(c) with composed virtual rooms (Application Level).

The basic concepts of this architectural application composition are described. For the development of a specific application, only some of these elements may be relevant or others may have to be added by the developer. In general, the tools should provide a firm starting point.

4.1. Multimedia Information Objects

In view of the design procedure, this is the final step and since for further understanding a more elaborate explanation is available[5].

4.2. Basic Working Environments

The working environments for applications are:

- **Content**
  Where a communicator is primarily concerned with the elements of information.

- **Style**
  Where a communicator is concerned with the projection attributes of information.

- **Spaceplace**
  Where a communicator is concerned with the spatial projection of information.

- **Timeline**
  Where a communicator is concerned with the temporal projection of information.

- **Webmap**
  Where a communicator is primarily concerned with the linking of information.

4.3. Functions

Script Level functionalities required for multimedia applications concern the subjects of:

- (a) presenting (b) browsing (c) editing (d) linking.

  - **Presenting** (consumer function)
    Concerned with making received information perceptible.

  - **Browsing** (consumer function)
    Concerned with cruising through received information according to attached links.

  - **Editing** (producer function)
    Concerned with modifying information that is to be transmitted.
Linking (producer function)
Concerned with referencing information that is to be transmitted.

4.4. Operational Pairs

The functions listed in Table 1. should behave conformingly across all multimedia applications.

Table 1. Functionalities with Operational Pairs

<table>
<thead>
<tr>
<th>OPERATIONAL PAIR</th>
<th>FUNCTIONALITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>(content; presenting)</td>
<td>present content, open content window</td>
</tr>
<tr>
<td>(content; browsing)</td>
<td>search for some specific piece of content</td>
</tr>
<tr>
<td>(content; editing)</td>
<td>select/paste, insert/delete</td>
</tr>
<tr>
<td>(content; linking)</td>
<td>use content link</td>
</tr>
<tr>
<td>(style; presenting)</td>
<td>present stylesheet, open style window</td>
</tr>
<tr>
<td>(style; browsing)</td>
<td>search for a style element</td>
</tr>
<tr>
<td>(style; editing)</td>
<td>insert/delete</td>
</tr>
<tr>
<td>(style; linking)</td>
<td>use style link</td>
</tr>
<tr>
<td>(spacelace ; presenting)</td>
<td>dimension spatial scaling, open spacelace window</td>
</tr>
<tr>
<td>(spaceplace ; browsing)</td>
<td>walk through contents at fix place or from place</td>
</tr>
<tr>
<td>(spaceplace ; editing)</td>
<td>translate, invert, scale, scroll/reformat</td>
</tr>
<tr>
<td>(timeline ; linking)</td>
<td>present time line moment</td>
</tr>
<tr>
<td>(timeline ; presenting)</td>
<td>alter presentation moment</td>
</tr>
<tr>
<td>(timeline ; editing)</td>
<td>use temporal link</td>
</tr>
<tr>
<td>(timeline ; linking)</td>
<td>present time line link</td>
</tr>
<tr>
<td>(webmap ; presenting)</td>
<td>search through a webmap</td>
</tr>
<tr>
<td>(webmap ; browsing)</td>
<td>add a content, style, spacelace, timeline, webmap link</td>
</tr>
<tr>
<td>(webmap ; editing)</td>
<td>switch to another webmap</td>
</tr>
</tbody>
</table>

4.5. Virtual Rooms

An application consists of several virtual rooms each of these can be compared with a window system within an application. This system consists of the actual window in which one or more media types are presented and functionalities which affect the presentation. For example, at one specific moment the tourist has two virtual rooms presented to him in parallel: one with a window for the road-map and a window for the photographs and another which consists of one window with audio and controls. Some functionalities may cover e.g. browsing through the photographs or turning the volume down.

The functionalities are realized through actions of a group of operational pairs, meaningfully sequenced in a script, i.e. some kind of algorithm. Information on algorithmic languages may be found in [3] which uses ESIEREL. There will be two types of virtual rooms:
(a) the read-only ones composed by the developer.
(b) the randomly changeable ones composed by the user.

How a specific application could be composed by a developer or a user is too specific in this context, for this would lead to statements like: this or that application is likely to contain a teleconferencing room, a movie-drome or a mailbox ( and other things). With respect to the model one general virtual room must be dealt with at this abstract level: the toolkit. It is composed by the developer and gives a user access to several management functions for virtual room development, thus making the method open in time. These functions can be performed on virtual rooms and operational pairs, provided. The toolkit should include the ability to: create, store, restore, delete, select, translate and activate an operational pair or a virtual room. The functionalities can be created as presented in Figure 3. Note that since each layer can rotate, e.g. content browsing and style editing are equally possible.

Figure 3. Architectural Application Composition

4.6. Fitting the Application Development Method

The power of the designed application development method will be exemplified now. A innovative multimedia example shows a person at home behind his MultiVision (some sort of TeleVision) who wants to make a trip to Mumbai, Pune. Therefore, he enters his virtual leisure room and taps in to a travelling agency's
information and reservation service. The person tells the system that he wants to visit Mumbai, Pune. Immediately a series of photographs about the city's well-known artifacts is shown. In parallel, a road-map is drawn displaying the route followed by the photographer and the National anthem is played on route, several hotels pass by and whenever he feels to, the client zooms in to one of these by requesting a video about its rooms and services. Now, the national anthem and the photo projection are stopped and the video is played. When pleased with a hotel, the client makes a reservation after having been informed about availability and prices of rooms. Then the National anthem is resumed accompanied by the photo series and the road-map routing. In the meantime, a bar pops up, offering tourist-guide information on paper or video however, pleased with what he has got, the client terminates the session.

Here, it can be seen that the total application consists of two major operational rooms to be called booth and hotel. These virtual rooms are presented as the sequence (booth; hotel; booth).

First booth opens the Mumbai presentation service and presents the photographs the matching road-map and the national anthem in parallel. Then, after the user interrupts to zoom in, hotel starts a video to present the selected hotel's rooms and services. Since the client is pleased with what he has got, he reserves some available rooms. Finally, booth resumes by playing the National anthem again and by showing more of Mumbai the tourist-guide which is offered next is ignored by the user who terminates the session.

Now suppose this user wants to go to a scientific congress on multimedia communications and that first he has to make a presentation to inform his superiors. Since the context in which the information will be reviewed is formal he may want to remove some distracting details from hotel: e.g. he may enter (content; editing) on video and browse across the timeline to the reserved hotel-room by means of (timeline; browsing) on video. After all information on other rooms and services is deleted the presentation of the remaining piece of Marathi video is switched to English via (style; presenting) on voice. To create a report conforming to the corporate layout the user finally rearranges the presentation windows by manipulating all information types with (spaceplace; presenting). The modified hotel may now be added to the user's virtual room pariport which will be transmitted for approval to his superiors. When these officials open pariport they may, for instance, browse directly to the summary to get an overview of the expected revenues and expenditures; if this information is not sufficiently convincing, the superiors may start processing the created virtual room pariport.

The example presented above, can only demonstrate some possibilities of applications which have been developed according to the designed method.

**Conclusion**

The multimedia objects represent real-world conceptual object to capture the reality of the world. It is very difficult to capture reality, the real-world model and method is usually application specific.

First, the fundamental considerations underlying the design of an application development method have been put forward. Then, step by step, the method has been designed by elaborating on a proposed set of architectural application components for multimedia communications. Finally, it has been applied in an example.

When the classification of a object is not certain, the object belongs to a set of classes with an associated certainty degree. Similarity measures should take this certainty degree into account during retrieval process.

**References**


**Books:**

01. Principles of Multimedia  
   By. Ranjan Parekh  
   Tata McGraw Hill Companies.

02. Hypertext and Hypermedia.  
   By. J. Nielsen  
   Academic Press.