NEW INTERACTIVE MULTIMEDIA APPLICATION MODEL USING OBJECTS STORAGE IN A DATABASE

HASAN HARASIS

Department of Electrical Engineering,
Faculty of Engineering Technology, Albalqa Applied University,
P.O. Box: 15008 Amman 11134 Jordan

Email: harasis.hasan@yahoo.com

ABSTRACT

This paper presents a new interactive multimedia application model which combines the advantages of object-oriented programming with aspect programming. The model proposes a structure for multimedia applications with a good reutilization of code in other applications and objects storage in databases. As a result, the global time spend for realization of an entire multimedia application is reduced and the size of the executing file is smaller.

Keywords: *Multimedia***,** *Database***,** *Object, Storage, Processin*g.

1. INTRODUCTION

In these days, it is very important to have an open environment for multimedia applications development, because there are many application developers with a good experience in this field, and this experience can be used for the functions library development.

From design point of view, the time necessary to realize a good concept and the modeling for application is much longer. From implementation point of view, the time for creation is in this case, much shorter. The global time spend for realization of an entire multimedia application is generally shorter. If it uses the real time content generation concept for pages, the time of physical implementations degreases very well.

For example, to realize a multimedia presentation for a company which produce hundred goods it is necessary to make hundred pages, one page for every product. If all the information about the product is stored in a database, the implementation task for all pages is to make the model for a single page, and after that to connect this page to database, it is not necessary to realize hundred different pages. In this dynamic way the reduction of the implementation time is considerable. More, the size of the executing file is smaller, because the object are loaded only when are used.

In this paper, a new interactive multimedia application model which combines the advantages of object-oriented programming with aspect programming is proposed. In this model, a structure for multimedia applications with a good reutilization of code in other applications and objects storage in databases is presented.

2. THE ARHITECTURE OF MULTIMEDIA APPLICATION MODEL

In the multimedia applications model architecture are four modules having a very strong interaction. Every module is composing by other small modules, and these small modules can be used very easy in other applications.

I use four categories of modules for the application kernel. The application kernel, generally is a specific part for a multimedia application, is not possible to reutilize the code, and for users is a close part (don’t accept modifications). This kernel is “the manager” of entire application. He uses the functions, procedure, routines which are stored in modules. In many cases, the module is the equivalent of a functions library. One of the kernel jobs is to verify and to validate all the new functions implemented by the users. The architectures modules categories are:

- Objects processing functions.
- User interactions.
- Input / Output Operations.
- Integrated technologies.

The Multimedia applications model architecture is presented in Figure 1.

2.1 Objects Processing Functions:

For Objects Processing Functions are two main categories:

– Temporal processing - the temporal aspect of objects is used in processing.
– Objects Processing - for this category are two types of objects processing:
▪ Single Object Processing
▪ Multiple Objects Processing.

In a multimedia application, we run in the same time different sounds, video sequences and animations. The applications run correctly when are free hardware resources. It is necessary to manage very well the application and the hardware resources; this can be realized by many temporal references. The Time-Line routines must to be very simple and easy to use (for a user with medium knowledge in the field of programming). The main jobs of the Time-Line routines are:

▪ Management for time constraint objects and for the hardware resources used for playing that.

▪ Verification if the hardware resources are free to use. If is not free for use, is necessary to stop playing the current object.

▪ Temporal references list construction for effects and transitions between objects.

▪ Management of temporal references jumping in collaboration with sub-modules GUI and Events.

Here, is useful to use the aspect programming techniques [1]. The very good results are achieved if are mixed the object oriented programming techniques for multimedia objects description and aspect programming techniques for temporal management of the events.

Object Processing Functions – for reducing the space that a multimedia application requires in the context of increasing the power of calculation, I propose the utilization of “on-line processing” concept. That suppose the processing of the objects is made during the running of application.

So, it can be used only one physical object in many apostasies (in several scenes it can wear a variety of forms - it can be scaled, it can be rotate or deformed, etc.) during the running of an application.

If the application uses a very large number of objects which suffers the same processing methods, handling this objects can be simplified by using the idea of processing directly the information and not store it.

We can meet two situations:

▪ The new generated object can be stored (the handler can personalize the multimedia objects starting from a model object).
▪ Or the object can be destroyed after utilization (if we need it again the object can be regenerated).

If the processes are very complex, there is the risk of losing a lot of time with calculation, and the result cannot be obtained instantaneous. The processing functions have a big diversity and depend of the element that is processed.

Sometimes its implementation can be very heavy. I recommend structuring of the elements, in a library that contains processing functions, which can be accessed later.

For users, can be allowed the personalization of these functions, this is realizable by modification of the processing parameters. The functions in this library use generally only one object during the processing part.

Transitions – during one multimedia application, you don’t know the moment in which the use decides to go to another scene of the application.

To realize a very interesting switch between scenes, frequently are used transitions or effects between objects.

When in those scenes are objects like videos and animations, those contain different positions in each moment, that’s why you cannot recalculate the transition between the two scenes (the transition between the objects from the two scenes).

The processing part it will be done in the moment decided by the user, when he wants the transition.

In this case it will be a process on-line, which will use the system resources where the application it running, and we will not talk about storing the results of the process.

For a new utilization of the processes (effects and transitions) I recommend the organization of a library with distinctive processing functions transitions. Is necessary for these functions, to exist a

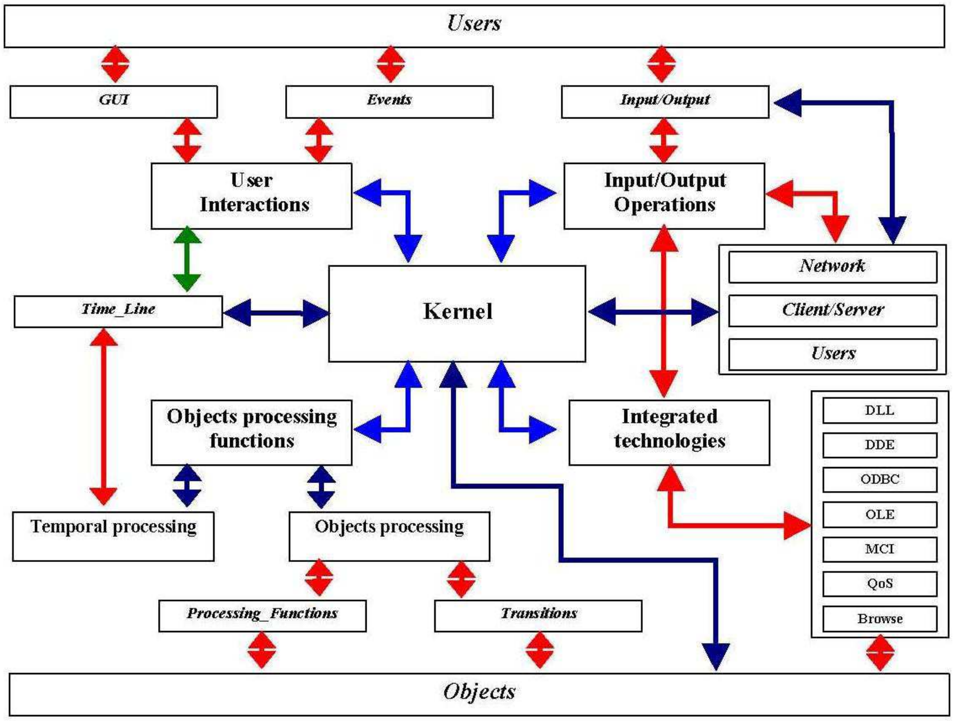

Figure 1 Multimedia Application Model Architecture

transparency of processing parameters, in this way the users succeed to personalize their applications.

Also is good that for experimented developers of multimedia applications to exist instructions on the way to implement new functions, which later can be distributed in new ways of multimedia applications area.

2.2 User Interaction

This module is characterized through a graphical interfaces and events generated from interactions between application and user.

GUI - The graphical interface with the user is made from different graphical elements, objects which offer interactivity between user and application (buttons, links, images, little animations, etc.). These objects have attached routine which start when one defined event appears in a library of functions, methods and events.

EVENTS - The events system modify the way of application run, depends on the actions made by the user in the application. This makes an event, and for them we will have a specified action. It is very important that we define correctly the system of events which might appear and also the actions which are realized after this.

It is necessary that we define one hierarchy of priorities in execution:

In case of mixing over the events in the same time;

In the case in which one event might provoke more than one action.

2.3 Input - Output Operations

Input - Output Operations contains also one component which permits the network work, the development of client/server and a system of user validation.

Input - Output Operations represents the way of interaction between the application and the out world (especially with the hardware resources of the computer system on which the application runs).

We can found these functions in all the programming languages. I propose the writing of small routines which can verify the peripheral state before their utilization, the release of some occupied resources because of some damage and the determination of peripheral performances.

In the relations with the functions from QoS library-checking services for the quality of the provided service – (the check for a good run of a multimedia application on a certain system) it's imperious the introduction of a routine for error detection that appears in the functionality.

Together with QoS functions there will be generated messages which will inform the user on the eventual problems that might appear and proposes the remove of problems.

Network – because now is a big variety of network types, and their development is continuous, we will check for the beginning the type of existing network, after that we will appeal to the specific routines for that types of network.

My recommendation, for the Network routines, is to have possibilities to verify the network level of traffic. Is necessary to inform the user about this level of traffic and to propose different solutions (if are some problems).

Client/Server - Many applications use for communication implementation of client/server services. These kinds of technologies are useful for job distribution in a heterogenic network, to enhance the speed of processing [2].

Users Management - If an application is accessed by many users in the same time, or sequential the routines for this module most to offer a management of users.

2.4 Integrated Technologies

This module contains libraries dedicated to working with databases (ODBC), integration libraries of the objects realized with other applications (OLE technologies), implementation of different kinds of services client-server libraries (DDE), with optimization functions for different equipments inside PC (MCI functions for sound, video and CD-ROM); evaluation libraries of quality service offered by the application on hardware resources by the user; function libraries which allowed searching object and resources which are used in the multimedia application; etc.

Dynamic Data Exchange (DDE) allowed the construction of client-server applications. DDE allows data actualization in other application and makes easier the utilization of remote commands.

Dynamic Link Library (DLL) use the extern files that contains routines and methods which are used by Windows applications.

Object Linking and Embedding (OLE) is a techniques which allows creation of objects and this objects can exported and reused in other multimedia applications.

Open Database Connectivity (ODBC) – Microsoft proposed the concept for generalizing the way of communications with databases.

I suggest a generalization in using this concept, so, the multimedia applications can use this way of connecting for accessing their objects from a database.

The ODBC technology allows a commune interface for accessing heterogenic SQL databases. SQL language is a standard of accessing databases. This interface gives a higher level of interoperability; one application can access a variety database system management which use SQL language. This allows users to realize and distribute a client-server application, without restrictions from a system databases management [3].

MCI – An operating system has to allow the use of multimedia resources for various applications which runs in the same time. Is necessary that their data stream to be synchronized, and for that you must use MCI functions.

QoS – Quality Services – This module describes the needs of a multimedia application from the point of view of components functionality [4].

I think that the model of a multimedia application has to contain evaluation elements of the hardware resource performance.

QoS module has to realize an initial testing of the equipments that is going to run. Is necessary to inform the user about how the application will run in these conditions (eventually it may propose solutions).

The biggest problems in multimedia applications are the synchronization constraints of different types of media, which runs simultaneous.

I propose that the next parameters which describe the quality of a service (of the way the program runs in a hardware environment) to be verified and respected by the programmer [5].

- The rate of application runs speed;
- The level of hardware utilization;
- The level of jitter.
- The temporal cumulate errors;
- The rate of environment errors;

Browse Functions – because multimedia applications are becoming more and more complexes (using more than 100,000 objects) is necessary to introduce a new module which contains browsing functions.

It is very difficult to go through the entire application to find information, that’s why; we prefer automatically solutions for searching. For this kind of applications is necessary to implement also content – based methods.

A correct structuring of the objects and also the implementation of the searching functions, offers many advantages to the users, increasing the attractiveness of the application.

The browse library has to be open for new development.

3. MULTIMEDIA OBJECT DESCRIPTIVE MODEL

Each multimedia object has a value, a logical structure and an interpretation of the content for that value. The value of a multimedia object, his logical structure and the model of representation and storages must be described in the multimedia object’s model.

The multimedia-describing model contains information that is referred to the logical composition of this object, the ways of synchronization and temporization for the components and the necessary parameters for displaying.

The model of interpretation has to describe exactly the real world through the sub-objects that are composing it and through the relationship between them and each sub-object and the real world.

One multimedia object may be composed with other multimedia objects or with simple elements (un-composed multimedia objects).

One multimedia object may be composing with the elements:

- Texts
- Images
- Sounds
- Videos
- Graphics
- Animations
- Scripts

The model of description can be synthesized like this:

- A number of simple or complex sub-objects compose each multimedia object, which is organized sequentially, parallel or in a mix mode.
- The elements from the object structure can take a value, which can define an instance of the object. It is unnecessary for the object to have defined values for all of its components. We use the heritage concept from object oriented programming.
- For a more real description of the real world it can be established a semantic interpretation for each component of a multimedia object.
- The semantic description became usefully in the case of searching different objects in a multimedia application or when the number of objects is very large. It is necessary to have a structure of the elements based on semantic concepts.

Next, I will present the parameters that are considered very important to me for describing a multimedia object and simple elements. I have organized object descriptions in a structure of classes that has close properties to the classes that we meet in the case of object - oriented programming.

A model of multimedia description is presented in Fig. 2

For an efficient description of multimedia objects I proposed a describing parameters structure:

General Parameters – that are describing the object, like object ID, storage physical location of the object, a flag which indicates if the object is simple or complex and information’s about element structure of the object.

Semantic Parameters – which describe the object through key words.

Type Objects Parameters - these parameters describe the object, its dimensions, the way of compression, the application that generated the object (useful for example in: the number of colors, dimensions, resolution, images, etc.).

Global Parameters – which describe the content of objects through many features. These parameters are useful in content database searching.

User Parameters – specified by each user and useful only in some applications.

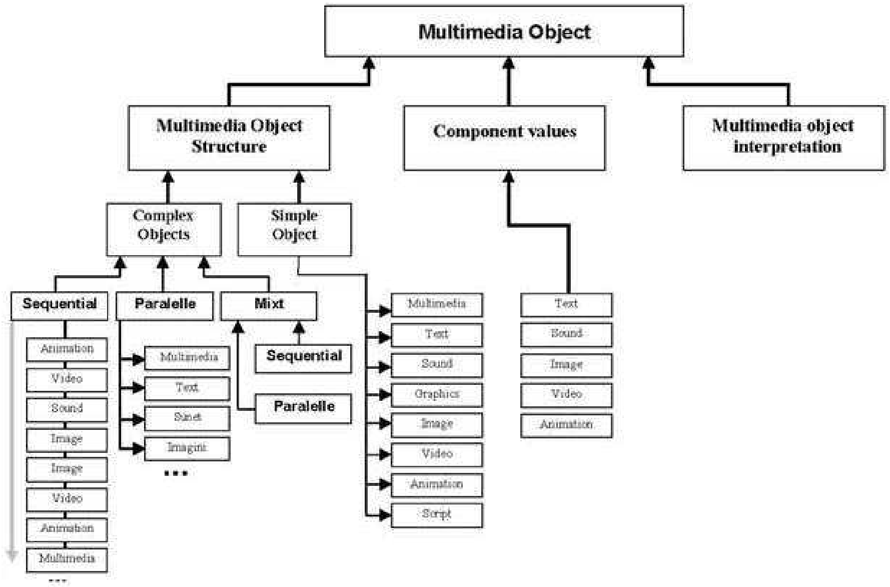

Figure 2: Multimedia Object Descriptive Model

After describing the object follows the description of the primary elements which are compose the object. The general parameters of description of the primary elements are:

Primary_Element

Parameters:

- Primary_Element Identification
- Primary_Element General Information's
- General Attributes

Primary_Element_Global_Features

Parameters:

- Primary_Element Description
- Specify Information's
- Attributes for Content Description of Object

Primary_Element_User_Features

Parameters:

- User ID
- Primary_Element Details
- User Personal Information
- User Attributes for Content Description of Object

4. THE MULTIMEDIA APPLICATION MODEL CHARACTERISTICS

The general characteristics of this model consist of:

- The possibility to reutilization of the elements developed previously;
- The possibility for any user to introduce new functions;
- Possibility to development an old application;
- Higher level of portability.

In concordance with the actual needs, my model present:

- Optimizations from the structure point of view, which is open for an ulterior development. I use for the primary level of structure modules. For all modules is possible to define new sub-modules. A sub-module accept functions, methods and routines, is possible to grow the number of that in the future (The user can add new functions, methods and routines).
- Is possible to reuse some parts of an old application.
- Object oriented implementation for structure of used objects in applications.
- Aspect oriented implementation for the temporal aspect of multimedia objects and applications.

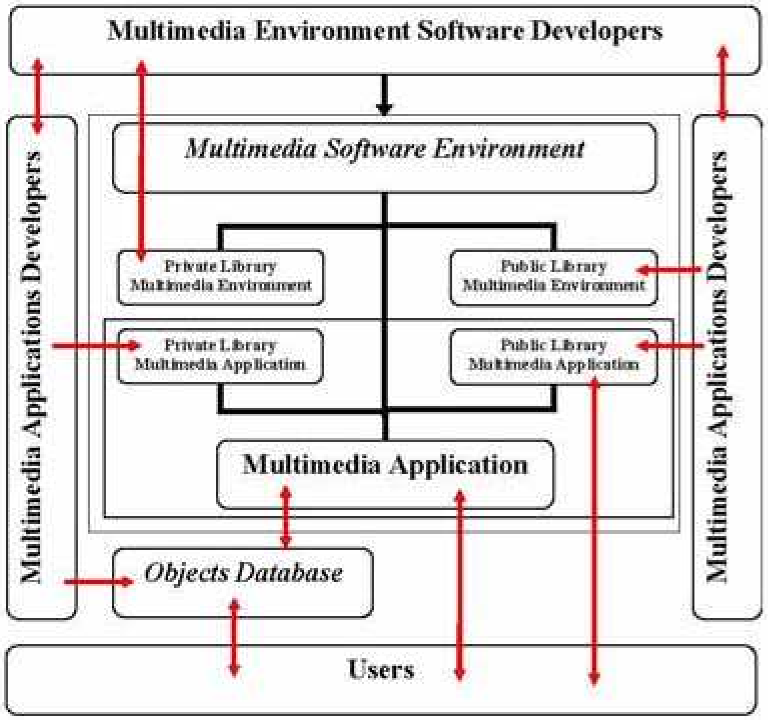
Multimedia Environment Software Developers
Multimedia Software Environment
Multimedia Applications Developers
Private Library Multimedia Environment
Public Library Multimedia Environment
Private Library Multimedia Application
Public Library Multimedia Application
Multimedia Application
Objects Database
Multimedia Applications Developers
Users

Figure 3: Multimedia Environment in the model context

- If the developers of multimedia environment use the proposed model, is possible for the users to develop the functions libraries and to transfer these between applications.

The model proposes a strong system for the quality of services. A special module QoS is developed for this.

- For the objects storage, can be use a standard database using a generalization of the ODBC concept.
- This model realizes a dynamic content for multimedia applications. The pages included in applications are generated dynamic from a database (in real time), this pages exist if the application runs, and are generated in concordance with the user options.
- Using the client/server technology, many users' access the application in the same time using remote commands.

I propose to include in every multimedia environment tools for creation of new functions library. In this way is possible to develop the multimedia software environment as shown in Figure 3.

.

5. CONCLUSIONS

In this work, we proposed an efficient interactive multimedia application model. This model combines the advantages of aspect programming and objects oriented programming. As a result, our proposed model applies a structure for multimedia applications that has good reutilization of code in other applications and objects storage in databases. In our proposed model, the time required for realization of an entire multimedia application was decreased and the size of the executing file became smaller. Using this model is possible to develop very efficient multimedia applications.

REFRENCES:

[1] Kiczales G., Hilsdale E., Hugunin J., Kersten M., Palm J., Griswold G. W. – "Getting Started with AspectJ", Raport of Defence Advanced Research Projects Agency F30602-C-0246, 2001

[2] Mocofan M., Pescaru D. – “Parallelisation of a textured colour image segmentation alghorithm”, Transactions on Automatic Control and Computer Science, Fourth International Conference on Technical Informatics - “CONTI 2000”, 12-13 October, Timişoara, Romania, pp 139 – 144, 2000.

[3] Khoshafian S., Baker B. – “Multimedia and Imaging Databases,” Ed. Morgan Kaufmann Publisher, San Francisco, California, 1996.

[4] Ghinea G., Thomas J. P. – “QoS Impact on User Perception and Understanding of Multimedia Clips”, Proceeding of The 6th ACM International Multimedia Conference, Bristol, England, 12-16 September, pp. 49-54, 1998.

[5] Mocofan M., Pescaru D. – “Aplicaţii interactive multimedia. Programarea OpenScript”, Ed. Politehnica, Timişoara, 2001.