ISSN 2047-3338

Model Designing for Multimedia and Hypermedia Application

R. N. Jugele and Dr. V. N. Chavan

***Abstract*– The new data modeling requirements have been arise mainly because of information technology improvements such as multimedia and hypermedia incorporation to documents. These and other considerations have produced new brands of evolving data models. Model based approaches to multimedia and hypermedia design are methodologies that make use of some of the newest model in order to begin from high level, a step by step process directed to low level specifications trying to cope with multimedia and hypermedia in a user oriented fashion. Representation of digitalized documents has considerable evolved with the integration of multimedia technology. Compared to textual documents, the content as well as the layout of multimedia documents has been extended to manage images, graphics, schemata and more generally any multimedia information. This paper presents a model dealing with multimedia information system using general development issues and specific multimedia subjects.**

***Index Terms*– Model, Real-Time, Multimedia, Hypermedia, Heterogeneous Applications, Distributed, Interaction, Retrospective, Synthesis and Systems Interconnection**

I. INTRODUCTION

FIRST restates that the paper is directed towards the design of an application development process which describes the model. To be able to define a model, we show its criteria and describe the omnipresent properties. Secondly, picturizes the model and deals with its environments. Finally, puts the model in perspective with respect to ISO's OSI and ODP models.

II. APPLICATION DEVELOPMENT MODEL CRITERIA

The designed model must fulfill several criteria which cover either general development issues or specific multimedia subjects.

A. Model Design Partitioning

Design partitioning is the action of composing an application development method such that the resulting application is well designed, i.e., it is characterized by flexibility, performance, and manageability [1]. The flexibility issue, covering suitability to change, involves complexity. This however opposes performance by decreasing throughput and increasing both runtime and response time. For the design this implies that the model should be kept simple by grouping types of applications and service classes and for the development this implies that Object-Oriented principles are to be used to conciliate these severely opposing characteristics. Finally, manageability within an application is achieved by incorporating security, reliability and maintainability for the model this brings along the demand for a structured stratification.

B. Retrospective Prospective

Retrospective is the intermediate system of Brown University [2], [3] and [4] is a framework for a collection of tools that allow to make links between standard types of multimedia documents created with heterogeneous applications.

Prospective introduces concepts for homogeneous application development (not the development of one homogeneous monolithic application). This will be based on a model that tries to unify analysis concepts introduced in [5], [6], and [7] with synthesis elements [15].

C. Overall Aim

The aim of the model is to serve in designing methodical tools to help the development of applications that run in a distributed and heterogeneous environment; at the same time, the model must serve in developing applications. As such it can be referred to as an application development model, as well as a multimedia communications model. The distributive and heterogeneity can be confined to geographical respectively technical dispersal of software and hardware data.

III. APPLICATION DEVELOPMENT MODEL DESIGN

Model should give the developer a universal structure which he can hold on to when analyzing or synthesizing any multimedia application [8]. A consequence of this ambition is that it should capture a variety of media, situations and possible applications. Yet, the model must be practical and usable too. The first requirement can be achieved by keeping the model simple and the second by focusing on core topics.

A. Basic Definitions

As per [16] the functional or a technical development environment is:

- Type of service (e.g., local; real-time)
- Type of interaction (e.g. user-user)

Before proceeding, some items may have to be explained the verb to communicate is used in its traditional Latinist sense, meaning to make become collective. Therefore, someone who uses an application can communicate either with another user (like in teleconferencing) or with a database (like in information retrieval). In addition, the term interaction is linked to the process of mutually dependent handling, the term functional is confined to actual functioning (not to functionality) and the term technical covers the subject of technological realization. In this view an object is a technical translation of a real-world entity.

B. Service Process

It takes place between communicators in equal or different spatiotemporal spaces. As an example, consider that the communicators act at different times then one may generate mail for the other. If the second communicator will be operational at a different place the mail will be either sent else, it will be left.

When temporal displacement is at stake, information transmission can be done in either real-time or store-&-forward fashion. If the communications can be realized in consecutive time-slices (i.e. one communicator acts after another) then the temporal relations between the objects are continuous and the service is typed as real-time else, in case of temporal dispersal, the temporal relations between the objects are synthetic and the service is identified as store-&-forward. This type of displacement may require e.g. token management or timeslot-assignment. In case of spatial displacement, the service can be either remote or local. If the communicators' applications have to take measures to correct presentation distortion due to geographical dispersal, then the service is remote else, it is local. This displacement may require:

- data synchronization at arrival due to intermediate delay or jitter
- realignment of data sequences due to overload
- error correction due to unwanted (deliberate or accidental) intermediate interference

So, for synchronization realignment and error correction the method that will be derived will not be transparent to location or migration.

C. Interaction Process

The interaction process between communicators is determined by the type of the creator of the content of the response on an interaction query. This cryptic expression may become clear when considering a user who is involved in a multimedia communications session. If the communicator interacts with is natural then the process is identified as user-user interaction else, (it is artificial) it is called user-database interaction. Teleconferencing (respectively information retrieval) has been the stereotype user-user (respectively user-database) interaction.

The reason for this distinction is that the interaction behavior of the communicators may severely affect traffic generation since bandwidth is the scarce commodity - this may affect implementability and implementation of an application for performance issues.

Consider the following example for the case of teleconferencing: suppose the communicators are driven by enthusiasm; then there may be more traffic generated than if the interaction was neutral. At the same time, the communicators may decide to interact through real-time (respectively store-&-forward) video which results in severely more (respectively less) traffic. Again, note that the ways an application will be used may affect its development.

D. Preliminary Summary

All the portrayed processes consist of components which are equivalent, discrete and disjoint. Applications classified in the model may resort under several components e.g. when some services are real-time and others not. To be able to generate a practical model it has been split in two major components, representing the functional and technical development environment.

IV. Model

The model in Fig. 1 consists of two three-dimensional cubes covering:

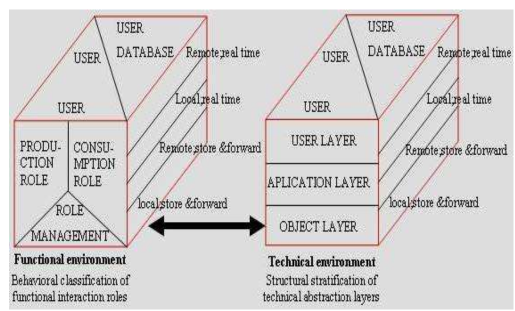

Fig. 1. Multimedia Communication Model

- the type of spatiotemporal service process which can be local or remote and real-time or store-&-forward (it identifies the interaction service between the communicators)
- the type of interaction process which can be user-user or user-database (it identifies the interaction relation between the communicators)
- the type of development environment which can be functional or technical

V. DEVELOPMENT ENVIRONMENTS

The model has been split in two major components, representing the functional and technical development environment the functional development environment will be identified as functional environment.

A. Functional Environment

This functional environment model gives an abstraction of the action behavior according to the assumed functional role of the application.

This environment deals with the access and interchange of complex multimedia information objects according to functional roles.

This part of the model, which is outside the scope of MHEG, covers three roles:

- the Production Role, where information is generated
- the Management Role, where information is transferred in orderly fashion between locations
- the Consumption Role, where information is received and presented

B. Technical Environment

This environment gives a hierarchically structured stratification in terms of technical abstractions for the realization of the objects. These abstractions cover the way multimedia information objects are built [9].

Here multimedia information objects become useful and accessible entities this environment of the model consists of three layers (which in total cover five sublayers, called levels). It is shown in Fig. 2 and explained below in order of increasing complexity.

At the User Level, there may be one or more users which are involved in interchanging information by means of the services of one or more applications; now, consider the situation where one application wants to communicate to another.

At the Application Level actual cooperation as seen by a user takes place through application-specific information interchange of e.g. feedback traces for information selection in a directory prior to retrieval.

At this level, communications are done by means of a script on the lower Script Level, a mere hyperlink based cooperation level which consists of structured scenarios specifying relationships within a set of multimedia information objects. This level includes procedures for e.g. navigation, external processor call or library function call. For this level, the de facto standards are Hypercard and ScriptX.

The previous levels use standardized objects at the Object Level i.e. the level the MHEG standard focuses upon. Here, structured multimedia information objects are related to the declaration of presentation entities describing e.g. composite Objects which include synchronization, input request objects, interobject relations, events and actions, links, object presentation (like audio volume) and mechanisms for referencing to the object content. In this area, MHEG competes with HyperODA, HyTime and MPEG System to become the industry standard [10].

The Object Level in turn accesses the Content Level which cares for presentation of basically unstructured data, represented according to monomedia standards for:

- text, e.g.. ASCII and Postscript
- graphics, e.g.. GKS and Postscript
- video, e.g.. JPEG (SPTV), or e.g., MPEG-video and DVI-RTV/PLV (MPTV)
- voice, e.g.. *G721n22* and PCM
- audio, e.g.. MPEG-audio and ADPCM

LAYER	LEVEL	FUNCTIONALITY
User layer	User level	User-specific multimedia communication
Application layer OSI application layer	Application level	Multiple multimedia application cooperation
	Script level	Scriptware multimedia application interaction
Object layer OSI presentation layer	Object level	Elementary MHEG inter-object interaction
	Content level	Monomedia content data description

Fig. 2. Technical Environment in Reference Model

VI. MODEL IN PERSPECTIVE

The designed application development model is related to both ODP (Open Distributed Processing) and OSI (Open Systems Interconnection).

Open Distributed Processing: As per ISO's ODP, the service process part of the model in Figure 1, may be classified in the engineering viewpoint (with respect to components of communications protocols) and the interaction process part may be compared with the computational viewpoint (with respect to interfacing). The functional environment viewpoint is omitted in ODP and the technical environment is identified as the information viewpoint (with respect to structuring aspects).

At the same time, the developed model prohibits location and migration transparency (in contrast with ODP) whilst concurrency and replication transparency are procured (in line with ODP).

Finally, like ODP, the developed model views a distributed system as consisting of parts which cooperate in order to provide users with the required functionality [11], [12] and [13].

Open Systems Interconnection: As per ISO's OSI, the technical environment model must in its entirety be put on top of OSI's session layer (layer 5). As indicated in Figure 2, the presentation layer in OSI is replaced by the Object Layer and OSI's application layer is then entirely substituted by the Application Layer which is used by the User Layer. In conformance with OSI, the layers in the application development model are separated by interfaces and a lower layer provides services to its higher layer only [14].

VII. CONCLUSION

The considerations underlying the design of a multimedia communications model for application development have been put forward. Then, step by step, the model has been composed by elaborating on the fundamental characteristics of multimedia communications. Finally, it has been compared with some globally accepted general development models.

By means of strictly defined set of interfaces the presented model allows the development of a core block on top of which applications can be easily built and plugged. Actually it is possible to find good, if not excellent, models supporting several aspects that compose the external and intermediate level.

REFERENCES

[1] B.G. West, Systems Design Techniques. The National Computing Center, 1984.
[2] N.K. Meyrowitz, The Architecture and Construction of an Object Oriented Hypermedia System and Application Framework. Proceedings OOPSLA 86. SIGPLAN Notices, September 1986, pp. 186-201.
[3] N. Yankelovich, N.K. Meyrowitz, and A. van Dam Reading and Writing the Electronic Book. IEEE Computer, Vol. 18, No. 10, October 1985, pp. 15-29,.
[4] N. Yankelovich, B.I. Haan, N.K. Meyrowitz, and S.M. Drucker Intermedia: the Concept and the Construction of a Seamless Information Environment.
[5] R. Steinmetz, Synchronization Properties in Multimedia Systems. IEEE Journal on Selected Areas in Communications, Vol. 8 No.3, April 1990, pp. 401-412.
[6] C. Nicolaou, An Architecture for Real-Time Multimedia Communication Systems. IEEE Iournal on Selected Areas in Communications, Vol. 8 (1990) No.3 (April), pp.391-400.
[7] T. D. C. Little and A. Ghafoor Synchronization and Storage Models for Multimedia Objects. IEEE Journal on Sel. A. in Communications Vol. 8 No.3, April 1990, pp. 413-427.
[8] R.N. Jugele and V.N. Chavan, OPEN ARCHITECTURE: Synthesizing Function to Interface with Real Time System, A National Journal of Science and Technology (Hislopia Journal), ISSN-0976-2124, Volume 3, Number 2, December 2010, pp. 221-234.
[9] R.N. Jugele and V.N. Chavan, Multimedia and Hypermedia Information Object: Object-Oriented MHEG Standard. International Journal of Computer Science and Telecommunications, Vol. 3, issue 6, June 2012.
[10] Various Multimedia and Hypermedia Information Retrieval Services: Investigation of Candidate Architectures for MHIR Services. ETSI DTR/TE-01017, September 1992.
[11] E. Moeller, A. Scheller and G. Schurmann Distributed Multimedia Information Handling. Computer Communications, Vol. 13 No.4, May 1990, pp. 232-242.
[12] L. B. Stefani, L. Hazard and F. Hom Computational Model for Distributed Multimedia Applications based on a Synchronous Programming Language. Computer Communications, Vol. 15 No.2, March 1992, pp. 114-128.
[13] I. Schultink, Modelling Service Creation in the Open Service Architecture. University of Twente (Master of Science Thesis), Iuly 1992.
[14] A. S. Tanenbaum, Computer Networks. Ellis Horwood, 1989.
[15] S. Gibbs, Composite Multimedia and Active Objects. Proceedings OOPSLA '91 (6-11 Oct. 1991 Phoenix, AZ, USA) Object-Oriented Programming Systems, Languages, and Applications. SIGPLAN Notices Vol. 26 No. 11, November 1991, pp. 97-112.
[16] R. N. Jugele and V. N. Chavan. Multimedia and Hypermedia: Communication Issues in Open Document Environment. Int. Journal of Computer Technology & Applications, ISSN: 2229-6093, Vol. 3(3), May 2012, pp.1043-1049.

R. N. Jugele received his B.Sc. degree from Nagpur University and M.Sc. degrees in Computer Science from Marathwada University, Aurangabad, Maharashtra, India in 1991 and 1993 respectively. Currently, he is working as an Associate Professor in Department of Computer Science, Science College, Congress Nagar, Nagpur, Maharashtra.

Dr. V. N. Chavan Head, Department of Computer Science, S. K. Porwal College, Kamptee, Dist: Nagpur. Maharashtra. He is research guide in Computer Science subject in various universities and has wide knowledge in research field since last 22 years. He is a member of various bodies in Computer Science subject.