New proposal for enhancing the interactive capability in MPEG-4

S. M. Tran1,2, M. Preda1, K. Fazeikas2 and F. J. Preteux1,
1 ARTEMIS Project Unit, Institut National des Télécommunications, Evry-France.
2 Budapest University of Technology and Economics, Department of Broadband Infocommunication Systems, Hungary.
Email: sony@mht.bme.hu.

Abstract—The paper gives a short overview on the new feature of the recent MPEG-4 standard, namely interactive facilitation of multimedia contents. Some experiments on dealing with BIFS and MPEG-J—the two main motors enabling the interactivity in MPEG-4—are presented. These experiments serve for two targets. First, they demonstrate a very perspective capability of MPEG-4 based multimedia contents, which are not only flexible in composition but also make interaction with viewer possible. Next, they expose a gap occurring in merging the computing and broadcasting technology in MPEG-4, namely temporal constraint versus random access. The work also outlines some solution to overcome this problem. The goal is to improve the performance of the interactivity enabling feature so that MPEG-4 standard actually integrates the computing capability into compression technique, enabling the generation of sophisticated multimedia scene.

I. INTRODUCTION

The boom of the Internet together with the rapid development in digital store capacity makes multimedia applications more and more popular. As a feedback, this new technology and related uses also expose some new requirements on the quality of multimedia applications. Among many others, consideration shall be put on a higher compression of image/video, the availability of composing different media data and the support of interactivity between users and applications. The emergence of the recent MPEG-4 standard brings efficient solutions to these new challenges. More than standardizing advanced tools for more efficient compression in the traditional fields (audio and video) of MPEG family, MPEG-4 jumps a great step forward with a new capability to build up a so-called multimedia scene, which can serve information to user in a much more innovative way than conventional movie: the interaction between multimedia content and users are now possible. With these new capabilities, MPEG-4 does put a steady milestone ahead in the convergence of the three worlds: TV/film/entertainment, computing and telecommunication. However, the advanced features of MPEG-4 have not yet been focused well enough in recent MPEG-4 applications. For instance, only the MPEG-4 audio video coding performance, having high quality at low cost of bitrate is exploited in DivX applications [15], in MPEG-4 capturing and/or playing [13], [14]. Advanced movie scenarios enabling composition and dynamic behaviors are still unknown or unfamiliar to most of video service providers. Correspondingly, researches on this field are still in the early phase with no feedback. Consequently, the MPEG-4 committee can hardly verify and improve these features of MPEG-4.

The research in this work targets at two objectives: exploring and improving the ability of composing multimedia content supported by MPEG-4. Toward these goals, we firstly introduce the powerful feature of MPEG-4 to generate a multimedia content with flexible composition and interactive behavior in the next section. Section 3 demonstrates several samples of such advanced multimedia contents at different complexity. The section also points out the need of temporal constraint, which is not straightforward in MPEG-4 and current broadcasting domain in general. Section 4 focuses on a possible solution for this shortcoming of MPEG-4. The paper closes with conclusion and perspective work in the last section.

II. OVERVIEW THE CORE TECHNOLOGIES FOR CONTENT COMPOSITION AND INTERACTIVITY IN MPEG-4

MPEG-4 addresses the coding of audio-visual objects of various types: natural video and audio objects as well as textures, text, 2D and 3D graphics, also synthetic music and sound effects. Even more complicated, there are possibly a large range of objects of the same type, which can be again categorized in numerous classes, based on their logical meaning. These classes can be repeatedly rearranged into subclasses, and so on. Encyclopedia application is such a good example of such complex multimedia content, challenging the capability of new multimedia standards including MPEG-4. To efficiently handle the huge volume of the information, the possibility of composition, reusing as well as manipulation on demand becomes crucial to multimedia compression. In MPEG-4, the incorporation of Binary Format for Scenes (BIFS) and Java technology are the key-techniques to satisfy the aforementioned requirements.

A. The parametric system with BIFS

In order to reconstruct a rich multimedia content—combining various types of audio-visual objects serving for different intentions—at client’s terminal, it is not sufficient to transmit only the compressed audio-visual data to that terminal. Additional spatial/temporal information is needed for enriched applications in order to combine media data at the terminal, to construct and to present to end-users a meaningful multimedia scene. The emergence of BIFS is dedicated to take over this role. This new type of information is a part of system
information, acting as an intermediate layer between media data (audio, still image, video) and the final built-up content. BIFS enables the modeling of any multimedia scenario as a scene graph where the nodes of the graph are media objects (Fig. 1a). MPEG-4 player no longer retrieves directly compressed audio / video data from transmission channel. Instead BIFS information is decoded first. Via the BIFS nodes appearing in the scene, decoder accesses the associated audio visual data as well as other (new) supplementary data defined in MPEG-4. Therefore, the presentation of the compressed audio / video data such as scheduling, coordinating in temporal and / or spatial domain, synchronization, processing interactivity, and so on can be flexibly controlled through the properties of BIFS nodes.

![Diagram](image.png)

Fig. 1: Different scenarios for video applications supported by the MPEG-4.

(a) The underlying structure of an enhanced video application based on MPEG-4.
(b) A snapshot of enhanced video application. Supplemental information: subtitle, logo graphics, arbitrary shaped image.

Similar to Virtual Reality Modeling Language (VRML) [4] [5], BIFS describes a scene with a hierarchical structure that can be represented as a graph. The structure of the graph is not necessarily static: nodes may be added, deleted or modified. In complicated scenario — interactive multimedia content — BIFS nodes can be dynamically added or removed from the scene graph on demand. Certain types of nodes — sensors — can interact with users and generate appropriate triggers, which are transmitted to other nodes by routing mechanism, causing changes in state of these receiving nodes. The effect results adaptive behaviors of the multimedia scene upon users' requests.

Scene (hierarchical structure of BIFS nodes) can be also designed to change over time. So a scene has a set of states similar to successive frames of a video. And in the purest MPEG tradition, BIFS scenes have initial states similar to video I (intra-coded) frames, and predictive states similar to video P (predicted-coded) frames. Scene I frames is packaged into an Access Unit (AU), which is a Random Access Point (RAP), and scene P frames are packaged into subsequent AUs, which are not RAPs. All the AUs in sequence build up the BIFS stream, which is multiplexed with the conventional compressed audio video streams to be sent to clients’ terminals.

B. The programmatic system with MPEG-J

The scene composition and interactivities provided by the BIFS information mentioned in section A represents a parametric description of multimedia content. Although the deployment of the so-called Script node in a scene also offers content creator a taste of “programming” the scene behavior within the scope of BIFS layer (the programming language here is the ECMA script [10]), the complex programmatic behavior in presentation of a multimedia scenario is realized with the integration of Java virtual machine at MPEG(-4) terminal. The vision behind MPEG-J is that an application written in Java is embedded in the MPEG-4 stream (it is called MPEGlet), so that it can flexibly dictate the reaction of compliant MPEG-J terminal in presenting the associated content in correspondence with user input and terminal conditions. As seen from Fig. 2, the controlling range of MPEG-J covers several units of the client terminal. When enabling interactivity and complex behavior by using BIFS script only interface to the scene graph is available; when using MPEG-J this interface is extended to the delivery and decoders layers. Therefore the MPEG-J maximizes the control capability of MPEG-4 content.

Java programming language is widespread and object oriented; it also provides — among many others — exception handling, event mechanism, polymorphism, platform independence as well as robustness and security. These were the main reasons that led to the choice of Java in MPEG. MPEGlets are required to implement the MPEGlet interface defined in the MPEG-J specification [1]. The MPEGlets and all other associated classes are carried in MPEG-J elementary stream, which is multiplexed with other MPEG-4 stream to be sent to client side. The Object Descriptors ODs (conveyed in their own elementary stream) determine the name scope of MPEG-J classes and objects as in the case of audio / visual objects.
III. DEMONSTRATING THE REQUIREMENT OF TEMPORAL CONSTRAINT

Generally, game-applications supply a high interactive scenario, which cooperate closely with user to help him / her to keep a long with the rules in the game. Therefore they create a good environment for verifying the computational feature of MPEG-4. In the rest of this section, we introduce two game-application created as MPEG-4 based “movies”: the Pinball and Memory game. They are a good starting point leading to the necessity of the temporal constraint in MPEG-4.

Fig. 3a captures a moment in the Memory game. Viewer (player) has to find all the matching pairs of image-cards, which are shown with their faces down. By two consecutive selects, he / she can turn them over. If the exposed images are the same, they will be removed. Otherwise the image-cards will be covered again and user shall remember their position for the other try. When all pairs of images are found and removed, the game becomes a conventional movie file: the main movie clip will be played.

Fig. 3b denotes a snapshot of the Pinball game. The target of the game is to keep the bouncing ball on the screen as long as possible. The deviation of the ball is restricted to 3 directions: left-, right side and top of the screen. At the bottom of the screen, player has a racket with a limited length to hit the ball back to the screen. To emphasize the movie-nature of the application, the texture of the ball is a miniaturized video clip, which will be played in full sized window when user successfully blocks the ball for a certain period of time.

Fig. 3a: Memory game
Fig. 3b: Pinball game

Both multimedia scenarios imply a strict temporal constraint, which can be violated in the streaming nature of MPEG (-4). Viewer can browse along the stream as in the case of traditional movie. The Compositor / Renderer (Fig. 2) then takes the last AU having RAP to recreate the game scenario. Considering the name scope of the object in the scene, the AU here belongs to either the BIFS or the OD stream (they respectively correspond to the parametric or programmatic method we use to create the game). It means that the
reconstructed scene will not take any state in the scene preceding the browsing operation due to the independent nature of RAP. In the above games situation, suppose that viewer has already won the game, and can see the disguised movie now; then he / she scrolls forward to skip the introduction part of film; it is impossible because the renderer will regenerate the game as if viewer had not ever won the game.

IV. PROPOSES SOLUTIONS FOR TEMPORAL CONSTRAINTS

The aforementioned shortcoming is due to the lack of data exchange between RAPs of BIFS stream as well as of the MPEG-J (actually the OD) stream. It can be considered as a shortage — on contrary, it is an advantage — in the conventional broadcasting domain. The more complexity of multimedia contents is available with MPEG-4, the less temporal dependency is required. However, the independency cannot simply be omitted, since the random access in broadcasting technology must be retained.

As a solution, we propose a modification in the behavior of MPEG-4 terminal while treating the BIFS and MPEG-J information. We define so-called static variables, referring to such variables (objects), whose name scopes stretch over the boundary of RAPs. If the previous RAPs are available to client’s terminal, the static variables will inherit their state. Otherwise, they are initialized as usual.

In the scope of BIFS information, the static variables can be implemented in two ways. Either the introduction of “static” keyword in declaring a variable in ECMA script or the definition of a new BIFS node can be solutions. In latter case, thank to the similarity in functionality, we name the new node after the Cookie conception in Hyper Text Marking Language technology. The syntax of Cookie node can be written in the fashion of Syntactic Description Language [1] as following:

Cookie{
    ExposedField MFString Name[]
    ExposedField MFString Value[]
}

In the scope of MPEG-J, the “static” type of variable is already defined. Therefore only their name scope must be extended farther than the boundary of RAPs of OD stream.

We are willing to introduce the persistent variables as well. The term persistent expresses the storability of some states, which may be useful for making bookmark, registering records, etc. in sophisticated multimedia scenarios. However, the security must be considerably taken into account to prevent any virus-like attacks toward client’s terminal.

V. CONCLUSION

MPEG-4 is the first standard, which enables the efficient manipulation upon a large scale of video / audio data. The standard presents a convergence between compressing methods for media data and programming languages for manipulating them. However, the programmable feature seems to be left aside in recent MPEG-4 implementations and applications, which mainly focused only on video / audio compression issues.

Our research here tends to reveal this unexploited feature of MPEG-4. A comprehensive study was made on the composition and interactivity power of the standard. We focus on the usage of MPEG-4 in creating multimedia game, the most complex multimedia scenario, which challenges back the capability of the MPEG-4. The problem, namely temporal constraint, is addressed and analyzed. The introduction of static state — one solution to the aforementioned problem — are discussed. Their deployments into the current structure of MPEG-4 terminal are also described. The actual implementation of the proposal in the reference software decoder of MPEG-4 and the introduction of the persistent state (another form of temporal constraint) are among our next targets in the future research.

REFERENCES

[10] ISO/IEC DIS 16262 Information technology - ECMAScript: A general purpose, cross-platform programming language
[12] C. Concolato, J. - C. Dufourd: Comparison of MPEG-4 BIFS and some other multimedia description languages, Workshop and Exhibition on MPEG-4, WEPM 2002, San Jose, California, USA.