EXPLORING MPEG-4 BIFS FEATURES FOR CREATING MULTIMEDIA GAMES

S. M. Tran[1,2], M. Preda[1], F. J. Preteux[1], and K. Fazekas[2]

[1] ARTEMIS Project Unit, Institut National des Télécommunications, Evry-France.
Email: sony@mht.bme.hu, Marius.Preda@int-evry.fr, Francoise.Preteux@int-evry.fr.
[2] Budapest University of Technology and Economics, Department of Broadband Infocommunication Systems, Hungary.
Email: kfazekas@cyberspace.mht.bme.hu.

ABSTRACT

This paper explores the new features of the recent MPEG-4 standard, namely the description and the composition of an interactive multimedia scene. Providing with such advanced facilities, the MPEG-4 standard is no longer considered only as an efficient compression tool for conventional video / audio application. It is now granted the power of computing mechanism as any other programming languages to facilitate the creation of high interactive multimedia contents. Game-applications therefore can be also considered as a new output-feature of the MPEG-4 standard. The work describes in detail how game can be created within the MPEG-4 framework. The process of making game with MPEG-4 is studied comprehensively to demonstrate the compatibility as well as the advantages of the MPEG-4 standard regarding to Flash technology, a well-known tool for making games over the Internet.

1. INTRODUCTION

The emergence of more and more powerful personal computers and pocket-devices together with the boom of the Internet expose diverse positive trends on the development of game-applications. They become more complex and at the same time more efficient. A large range of games is available. Games exist in all the operating systems of computers; they extend from a very simple game with straightforward graphic view, up to a sophisticated one equipped with a complex 3D environment; they can be played locally or collaboratively over a network,…The successful era of computer game we witness today is a result of a drastic improvement and increasing extension in developing environments offered to game-makers. The recent success Web technology of Flash from Macromedia Inc. can be considered as such an attribution to this progress. Flash originally provides a solution for composing multimedia contents over the Internet. But developers successfully exploit it to create plenty of Web game-applications with light load at clients' terminals [10].

Another development also promoted by the Web technology is the MPEG-4 standard, a sophisticated video / audio compression technologies. As a member of the MPEG family, the MPEG-4 standard inherits and improves all the advantages of its predecessors with graceful key-techniques, such as advanced audio coding, object-based video compression, deployment of wavelet-based texture encoding and mesh-based representation. But the most advanced feature of MPEG-4 is probably the concept of the multimedia scene supported by the definition of BInary Format for Scenes (BIFS), which acts as an intermediate layer between media data and the final built-up content. BIFS provides a flexible way of scheduling, temporally / spatially coordinating and synchronizing various types of media to create a so-called multimedia scene – a new concept, which replaces the conventional layout of movie, simply having one video and one audio at a time. Also with BIFS, MPEG-4 - a native compression technology of media data – is now enhanced with the computing power, a natural property of programming language. Subsequently, we consider it as a new competent tool for game-developers. The investigation of this additional capacity of MPEG-4 in this paper is structured as follows. The next section presents a brief review of BIFS structure and functionality. Section 3 describes how MPEG-4 BIFS can be involved into process of game creation. In Section 4, MPEG-4 BIFS versus Flash technology is addressed in a comparative discussion in terms of nodes / classes and functionality. Finally, Section 5 concludes the paper, outlining new perspectives for developments of games with MPEG-4 BIFS as well as pointing out our related researches in the future.

2. OVERVIEW OF THE MPEG-4 BIFS

MPEG-4 addresses the coding of audio-visual objects of various types: natural video and audio objects as well as textures, text, synthetic 2D and 3D graphics, also synthetic music and sound effects. In order to render this rich multimedia content at a client's terminal, it is not sufficient to transmit only the compressed audio-visual data to that terminal. However, this is the most common case in the current MPEG-4 implementations [8], [9]. Additional information is needed for enriched applications in order to combine various types of media data at terminal, to render to end user a meaningful multimedia

scene. BIFS is dedicated to take over this role. Several profiles of BIFS [1] are defined to smoothly introduce BIFS information into conventional video applications. With *Simple 2D* Scene Graph and Graphic Profile, a scene typically contains one video and one audio, which can be played at a time (Figure 1) similarly to the scenario offered by MPEG-1 / 2. According to the current development of players [7], *Advanced 2D* scene can be viewed, in which most of the 2D objects addressed by MPEG-4 are implemented. Players enabling higher profile will be available soon, to demonstrate the enriched combination of media objects supported by MPEG-4.

Similar to Virtual Reality Modeling Language (VRML) [2], BIFS describes a scene with a hierarchical structure that can be represented as a graph (Figure 1). Nodes of the graph build up various types of objects, such as audio video, image, graphic, text, *etc*. Thanks to hundreds of BIFS nodes standardized in MPEG-4, a large scale of scene from the simplest content with one audio and one video up to a sophisticated 3D content can be constructed. This resource is increasingly extended by the MPEG-4 workgroup. To ensure the flexibility, a new, user-defined type of node derived from a parent one can also be defined on demand with *Proto* method as in VRML. In Figure 1, instead of taking data directly from a bitstream, sending them to proper decoders and rendering the decoded data to displaying devices (speakers, monitor) in a synchronized manner as conventional movie-players, the BIFS information must be retrieved first from the same multiplexed stream and then decoded by a BIFS decoder. Some so-called media nodes, for instance AudioClip MovieTexture having underlying audio / video, involve the associated data from appropriate decoders whenever they are activated (dash-arrow in Figure 1). Other nodes such as Text, Circle can operate independently of the compressed data. In general, nodes expose a set of parameters, through which aspects of their appearance and behavior can be controlled. By setting these values, scene-designers now have a tool to force a scene-reconstruction at clients' terminals to adhere to their intention in a predefined manner. In more complicated scenario, the structure of BIFS nodes is not necessarily static; nodes can be added or removed from the scene graph arbitrarily. Certain types of nodes called sensors, such as TimeSensor, TouchSensor, can interact with users and generate appropriate triggers, which are transmitted to others by routing mechanism, causing changes in state of these receiving nodes. They are bases for the dynamic behavior of a multimedia content supported by MPEG-4. The maximum flexibility in the programmable feature of MPEG-4 scene is carried out with Script node. By routing mechanism to Event In i attribute of Script node, the associated function (defined in its *url* attribute) with the same name Event In i() will be triggered. The behavior of this function is user-defined, *i.e.* scene-designer can freely process some computations, then sets the values for every Event Out j attribute, which consecutively effect the states of other nodes linked to them. Direct manipulation of nodes' states is also available: the Field I attribute can refer to any node in the scene; through this link, all attributes of the contacted node will be exposed to direct setting and modifying operators within the Script node. The syntax of the language used to implement the function of Script node is ECMA [5]. With the routing mechanism and the programmability of Script node, MPEG-4 now claims a real power of computing tool. It is the reason why we believe that it is a competent developing tool for game-application in low bit-rate channel such as the Internet.

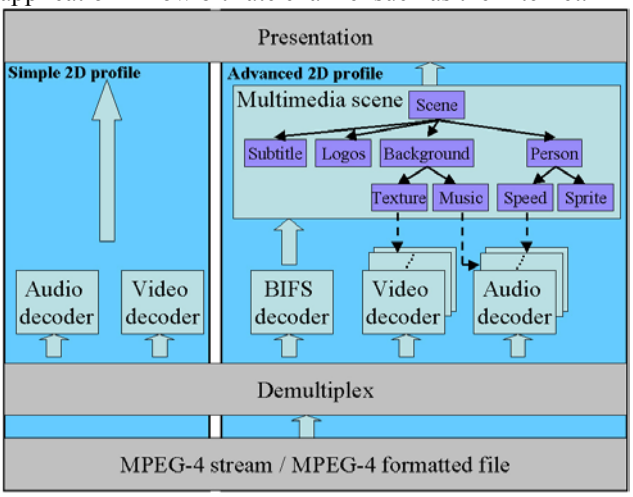

Figure 1: Conventional versus BIFS enabling scenarios.

3. MPEG-4 BIFS FOR CREATING GAMES

Generally, game-application supplies a high interactive environment cooperating closely with user to help him / her to achieve a predefined target. But game-application can also be considered as an instance of a complex collaborative multimedia content, whose composition is now possible within the MPEG-4 framework. Figure 2 shows a "movie" of Pinball game, a well-known computer application. Its "executable" storage is MPEG-4 formatted file (*.mp4*), which is more than a format for conventional movie. We use MPEG-4 BIFS as a "programming language" to create this game. Any MPEG-4 player software, which is compliant to *Advanced 2D* profile scene graph, supports the environment to run the Pinball game. Currently, we use EnvivioTV [7] to test the game on 256MByte RAM, Pentium III 800MHz computer. For game-applications involving more than two MPEG-4 video decoders at a time, higher configuration of computer is necessary to produce acceptable effects of games.

The rule of the Pinball 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- and top side 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. We outline the essential tree-graphs of scene-objects in Figure 3. Their properties / behaviors can be summarized as follows:

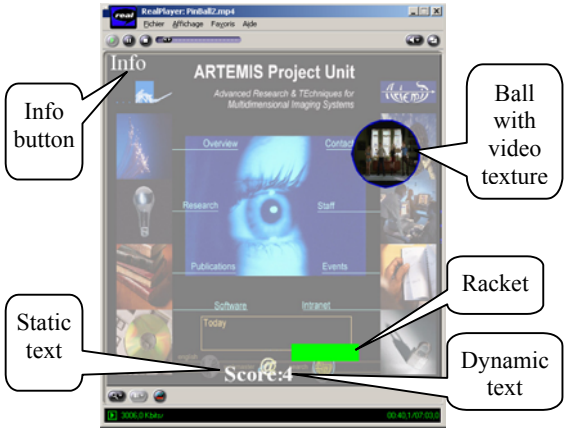

Figure 2: Pinball game with MPEG-4 BIFS.

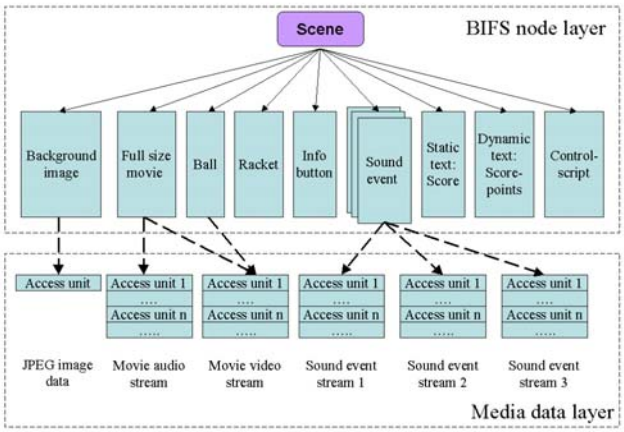

Figure 3: MPEG-4 scene-structure associated with Pinball game.

1. Some objects refer to underlying media data (*Background image*, *Full size movie, Ball* and *Sound event*). They are responsible for displaying an image / playing a video clip, a sound event (for instance, sound effect is played when ball is hit by racket) and an audio clip. These objects address the associated media data through several types of media nodes such as ImageTexture, MovieTexture, AudioClip or AudioBuffer.
2. Some objects are passive (*Static* / *Dynamic text*): their operations are independent of the media data layer. They enhance the scene with decorative graphic shape or useful information. Their appearance can be constant or changeable during the lifetime of multimedia content. In our case, the *Static text* displays the "Score" text, while the *Dynamic text* shows the current score-points of player, which is changed time to time.
3. Some objects are sensitive (*Racket*, *Info button*). They contain at least one sensor node: PlaneSensor2D and / or TouchSensor. These nodes inform the scene-designer about users' interactions through the mechanism of signaling, giving him / her hints to react accordingly. In our case, we route the translation_changed attribute of PlaneSensor2D and the isActive attribute of TouchSensor to *Control-script* object to capture respectively the movement of *Racket* and clicking action on *Info button*.
4. There is a controlling object (*Control-script*). This object plays a key-functionality in the logical behavior of the whole multimedia content. It contains one or more Script node; each has some Event In attributes, which are targets for several routes from other nodes. The associated function will process the predefined computation whenever the connected route is triggered. For instance, as a scene-designer, we want to display a *Static text*, which instructs player the rule of the game, when he / she clicks on *Info button*; the isActive attribute of the TouchSensor node (component-node of *Info button*) is routed to the InfoClick attribute of the Script node in *Control-script*; In the InfoClick () function of this Script, using syntax in [2], we set the transparency attribute of *Static text* (this attribute belongs to a Material2D node, a component of *Static text*) to zero, that is visible. This value is originally equal to 1, forcing *Static text* invisible as default.

4. COMPARISON OF MPEG-4 BIFS VERSUS FLASH TECHNOLOGY

Macromedia Inc. is a well-known software company, which develops about 20 products for the Web. Among them, the tool that interests us is Flash, which is currently used to implement a lot of Web-distributed games [10]. Flash uses a proprietary binary format for publishing its application over the Internet called Swiff (*.swf*). Though propriety, the specification of Swiff and its software developer's kit (SDK) are public at the Web site of Macromedia Inc. [6]. It can be seen from SDK that the file is structured into high- and low level manager. Classes in the former level are closer to logical objects built directly above the SDK (objects in an authoring tool), while those in the latter straightly expose the physical structure in Swiff file. Some possible mapping between classes in the high level manager and BIFS nodes of MPEG-4 can be made as in Table 1. Therefore we can assume that Flash-based games can also be implemented with MPEG-4 BIFS. Then numerous advantages offered by MPEG-4 will be inherited in the game-application. We refer readers to [3] for complete list of features, exploring the comparisons between the 2 technologies. We highlight the following main points:

- MPEG-4 is a native audio-video compression. It claims an efficient storage and transmission these media data. It is possible to make the synchronization for several media on a per frame basis.
- MPEG-4 in general provides hooks for digital right management, which may be useful for doing business with game-applications.
- Not restricted to IP network (the Web), MPEG-4 content can be routed in an MPEG-2 transport stream,

which is designed for transmission over various scenarios of media channels.

- Similar to other international standards, MPEG-4 will be widely supported by hardware-manufactures as well as third-party software developers. Therefore, games based on MPEG-4 are expected to spread to large range of terminals such as cellular phones and personal digital assistants (PDA).

Classes in high level manager of Swiff file	MPEG-4 BIFS structure
HFMovie	SFTopNode: Group, Layer2D,…
HFFrame	Temporal BIFS commands construct scene at a time.
HFObject	SFWorldNode: general collection of nodes.
HFSound	SFAudioNode: AudioBuffer, Sound2D,…
HFButton	SFGeometryNode (Circle, Rectangle,…)+ SensorNode
HFShape	SFGeometryNode: Circle, Rectangle,…
HFRectangle	Rectangle
HFBitmap	SFTexture: ImageTexture, MovieTexture,…
HFCircle	Circle
HFPolygon	IndexedFaceSet, IndexLineSet,
HFOval	IndexedFaceSet, PointSet,…
HFFont	FonStyle
HFText,	Text
HFEditText	Text
HFAction	Script

Table 1: Compatibility between MPEG-4 BIFS and high level manager classes in Flash.

It is also worth mentioning some obstacles one shall meet currently while developing game-applications / multimedia contents with MPEG-4 BIFS. They can be summarized as follows:

- There is almost a few authoring tool for MPEG-4 BIFS. The authoring tools often require scene-designer a deep knowledge of BIFS structures.
- The integration between BIFS node layer and Web browser does not exist explicitly as in Flash and VRML. The current possible solution is going through a heavy layer addressing Java technology: J-MPEG stream also supported in MPEG-4 [1].
- MPEG-4 players supporting 3D scene are still under development, partial complete. The full advantage of MPEG-4 is only available in the near future.
- Semantically, BIFS layer does not support storing the state of a multimedia scene for latter use especially in case of game-application. The state can be "static" for its whole playing time or "persistent" for any later replaying time. Because of the movie nature, player can use the external controller to scroll backward or forward. Without "static" variables, chronological events of the game may be violated. With "persistent" variables, for instance, high score of game can be stored for the next playing.

5. CONCLUSION AND PERSPECTIVES

MPEG-4 is the first open standard, which enables the scene-related information – BIFS – transmitted together with video / audio data. The functionality of BIFS shortens the gap between conventional movies and interactive computer program. Our work here shows that game-applications can be developed with the power of this unexploited feature of MPEG-4. Even more encouraging, we succeed in outlining the compatibility between MPEG-4 BIFS and Flash technology, a platform widely used for creating game on the Internet. Games based on Flash do not make a high load at clients' terminals. It is the result for its widespread against Java applet-based although Java technology possesses a real powerful programming language. But with MPEG-4 BIFS, Flash may face a considerate competitor. The fact is true for Flash and also for all of other programming techniques that they may claim a powerful computing capacity, but none of them deal with compression of audio, video as MPEG-4 does. Therefore when enhanced with a computing power thanks to BIFS, MPEG-4 can result in a light load but complex game application for the Internet communication. Furthermore, 3D nodes in VRML are adopted in MPEG-4 BIFS, we can expect efficient online 3D games in the near future.

Developing a high abstraction level of authoring tool for MPEG-4 BIFS, creating an automatically converting tool to generate MPEG-4 -based application from Flash-based ones are main targets of our future work. Figuring out the solution for "static" and "persistent" variables as well as contributing to the implementation of 3D MPEG-4 player are also scheduled as our research perspectives.

6. REFERENCES

[1] ISO/IEC 14496-1:2001.1 *Information technology coding of audio - visual objects: system part*, 2002 March.

[2] ISO/IEC 14772-1: 1998, *Information technology — Computer graphics and image processing — The Virtual Reality Modeling Language — Part 1: Functional specification and UTF-8 encoding*.

[3] 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.

[4] F. Pereira, T Ebrahimi, *The MPEG-4 book*, IMSC Press Multimedia Series/Andrew Tescher, 2002.

[5] ISO/IEC DIS 16262 *Information technology - ECMAScript: A general purpose, cross-platform programming language*.

[6] Macromedia, Inc., http://www.macromedia.com.

[7] Website of Envivio, http://www.envivio.com.

[8] Website of Philips, http://www.mpeg-4.philips.com.

[9] Website of DivX, http://www.divx.com.

[10] Gallery of Flash-based games, http://www.flash-game.net