Eurographics Italian Chapter Conference (2006)
G. Gallo and S. Battiato and F. Stanco (Editors)

Using Layout Profiles in MPEG-4 Hypervideo Presentation

A. Lo Bue

ICAR-CNR Palermo, Italy

Abstract

The definition of presentation layout, based on device and user features, can be useful to provide multi-devices content adaptation capabilities for hypervideos presentations. We provide a possible approach of content adaptation through different devices using MPEG-4 as presentation framework. In particular, we focus on the use of adaptable interfaces in hypervideos presentations, separating layout information from media data. A possibility of adaptation comes from separation of media representation and audiovisual scene structure in MPEG-4. We describe an implementation of this approach through XMT-A language, defining two layout examples and denoting advantages and problems encountered.

Categories and Subject Descriptors (according to ACM CCS): I.3.6 [Computer Graphics]: Device independence, Standards

1 Introduction

1.1 Background of MPEG-4

MPEG-4 is an ISO/IEC standard developed by MPEG (Moving Picture Experts Group). MPEG-4 provides a framework for description, compression, storage and transmission of audiovisual data. These audiovisual data are called *Media Objects* (*MOs*). *MOs* can be of natural or synthetic origin; MPEG-4 describes the composition of these objects to create compound *MOs* that form audiovisual scenes; it also multiplexes and synchronizes the data associated with *MOs* and defines interactions with the audiovisual scene generated at the receiver's end.

MPEG-4 audiovisual scenes are composed of several *MOs*, organized in a hierarchical fashion. MPEG-4 defines the coded representation of such objects. A *MO,* in its coded form, consists of descriptive elements that allow handling the object in an audiovisual scene.

In addition to providing support for coding individual objects, MPEG-4 also provides facilities to compose a set of such objects into the scene. The necessary composition information forms the scene description, which is coded and transmitted together with *MOs*. Starting from VRML (the Virtual Reality Modelling Language), MPEG has developed a binary language for scene description called BIFS (BInary Format for Scenes). In order to facilitate the development of authoring, editing and interaction tools, scene descriptions are coded independently from the audio-visual media that form part of the scene. BIFS is a compact binary format representing a pre-defined set of *MOs*, their behaviours, and their spatio-temporal relationships. BIFS scene description in general can be time varying. BIFS data are carried in a dedicated *Elementary Stream* that contains the coded representation of either audio or visual data, or scene description information or user interaction data.

MOs may need streams of associated data, which are inserted in one or more *Elementary Streams*. An *Object Descriptor* identifies streams associated to a *MO*, allowing the hierarchically use of all encoded data. *Elementary Stream* descriptors include information about the source of the stream data, in the form of a unique numeric identifier (the *Elementary Stream* ID) or an URL pointing to a remote source for the stream.

The MPEG-4 standard also defines a storage file format. The MP4 file format is designed to contain the media information of an MPEG-4 presentation in a flexible, extensible format, which facilitates interchange, management, editing, and presentation of the media.

Representing MPEG-4 scene description using a textual syntax is possible using the Extensible MPEG-4 Textual format (XMT) framework. The XMT framework consists of two levels of textual syntax and semantics: the XMT-A format and the XMT-Ω. While the XMT-Ω is a high-level metadata abstraction of BIFS, the XMT-A is an XML-based version of MPEG-4 content, containing a subset of the X3D and providing a one-to-one mapping between the textual and binary formats. The XMT-A language contains a set of descriptors, which allow *Elementary Streams* identification, description and association with others scene objects. An XMT representation can be compiled and stored into MP4 format, and thus played from a MPEG-4 player.

In our work, MPEG-4 scene representation metadata are written in XMT-A. The scene description contains spatio-temporal relations between audio, video, synthetic and textual objects.

1.2 XMT-A and BIFS

An XMT-A document instance is composed by a set of elements to represent MPEG-4 systems streams as a textual format. In the textual format, the order of the elements in

the document is not necessarily the same as in the corresponding binary constructs in the streams. An XMT-A document has a single optional <Header> element followed by a single <Body> element as in figure 1.

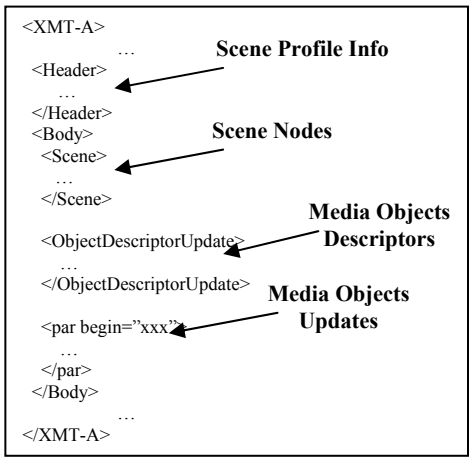

Figure 1: *XMT-A document nodes structure.*

The BIFS scene description consists of a collection of nodes that describe the scene structure. One or more nodes describe an audio-visual object in the scene, which may be grouped together (using a grouping node). *Elementary Streams* can be of different type, based on carried data. A single audiovisual scene is usually conveyed in a *SceneDescription* type *Elementary Stream* (usually called BIFS stream). However, multiple scene description data may be conveyed in more than one BIFS stream, using <Inline> nodes in a BIFS scene description where each such node refers to further BIFS streams. In this case, multiple BIFS streams have a hierarchical dependency. Each <Inline> node opens a new name scope for the identifiers used to label BIFS elements.

1.3 Related Work

Hypervideo is the next generation of video (in literature Time-based Hypermedia), a kind of interactive video that connects different video sequences to create dynamic narrative stories that change story evolution as the user makes choices selecting links between the scenes. Hypervideo definition is encountered first time in [SBI96] as digital video and hypertext, offering to its user and author the richness of multiple narratives, even multiple means of structuring narrative (or non-narrative) with links opportunities of spatial and temporal type. Hypervideo was also defined in [RVM02] as a sub-class of hypermedia based on the link-node structure of hypertext, and composed of digital video media. Like static hypermedia, more than one opportunity can be presented at once, but unlike hypertext, these opportunities are active only for certain time.
The opportunities are presented in the form of hyperlinks, which can be of two types: temporal and spatio-temporal. Temporal links are exhibited in the form of annotations that can appear on top of the video during a certain period. They can also be presented in form of previews of destination video scenes that are played back for a specified duration, at specified points in time during the playback of the source video scene. Spatio-temporal links take advantage of the hierarchical structure of the video and associate opportunities to particular objects at a certain time.

The use of hypervideo presentations is the present and the future of video communications. In particular, hypervideo presentations are useful for cultural heritage applications, video learning systems and interactive TV, giving to users a high interactive, attractive video experience. In [MTLT05] and [LoB05], the authors describe a methodology for authoring MPEG-4 hypervideos using *video objects* as spatio-temporal link opportunities. Authors use *video object* concept as a collection of regions exhibiting consistency across several groups of contiguous frames, in at least one semantic feature. They propose to describe *video objects* with MPEG-7 descriptors and subsequently map these descriptions in XMT-A metadata structures as synthetic objects through XSLT.

Adaptive and adaptable interface authoring, based on multi-device layout adaptation, was inspected in some works from different points of view. In our work, we propose an adaptable authoring approach. Looking for semantics of both terms, Bulterman et al. define adaptive presentations as presentations capable of automatic adaptation, compared to adaptable presentations, which requires external intervention to be adapted [BRHO99]. In [GZ02] Grundy and Zou propose an architecture for building adaptive user interface. In their work, the authors describe AUIT technology, which aims to provide an abstraction language for layout design, using a set of device independent XML tags that at run-time are mapped to HTML or WML interfaces. Another method was followed by Vanderdonckt et al. in [VLFO*01] which propose a modulator that intelligently transform a given user interface (UI) from one context to another one, allowing designers to build UIs across different platforms. In respect to this two approaches, we do not use any abstraction language or modulator, and thus we exploit the versatility of the MPEG-4 standard, without requiring a new authoring language definition.

The rest of paper is organized as follows: Section 2 presents a work overview. Section 3 presents the logical structure of a hypervideo example used as case study. Section 4 describes layout profiles definition for MPEG-4 hypervideos. Section 5 summarizes our work and gives an outlook of future work.

2 Work overview

We propose a methodology to separate layout information from media data of a hypervideo presentation. Hypervideo scene layouts are encoded using a different MP4 file for each profile. Involved media are linked from other MP4 files and adapted at presentation time to the selected layout profile. This approach permits to a hypervideo presentation to be adapted for specific devices and user choices. From interfaces point of view, hypervideo visualization has many

variables and content organization choices: user attention in hypervideo is based on presentation interface appearance; the device type used for hypervideo presentation influences content adaptability categories. In this scenario, the definition of presentation layouts can be useful. These layouts are based on interface organization and structure, to which hypervideo content presentation can be adapted.

One chance of adaptation comes from separation of MPEG-4 media representations from the audiovisual scene structure. This allows adaptability through layout profiles. Hypervideo, described through the XMT-A language, is organized from a logic point of view in *channels*. A *channel* is a virtual display of playback device like a window or a frame, able to play a media file, which can be used by one medium at time [Gag03].

The MPEG-4 scene composition (BIFS) language allows multiple hierarchical audiovisual scenes. We use this opportunity to separate hypervideo layout from *MOs* within the scene. In our work, scene layout components are represented in a XMT-A document containing graphical primitives and *channels* spatial positions. All encoding and representation info of media composing the hypervideo are instead written and encoded in other XMT-A documents (one for each *MO*). Media XMT-A scenes are linked to the main scene (containing layout info) and subsequently encoded in MP4 file format, allowing an MPEG-4 player to visualize the hypervideo presentation.

3 Hypervideo structure

In our hypervideo example, the audiovisual scene is composed of the following *MOs*: one image, used as interactive map for selecting hypervideo *narrative units*, one video sequence, and one textual caption associated to video sequences (updated for each *narrative unit* context). Video sequences are composed of two scenes linked through a spatio-temporal link opportunity (using *video objects* defined in [MTLT05]). These linked scenes constitute a *narrative unit* of the hypervideo. Finally, the hypervideo story is composed of many *narrative units* selectable from the interactive map (figure 2).

From a logical point of view, in our example, the audiovisual scene contains three *channels* used for video, text captions and image. The video *channel* plays narrative units selected from the map, while text *channel* contains captions relative to *narrative units*.

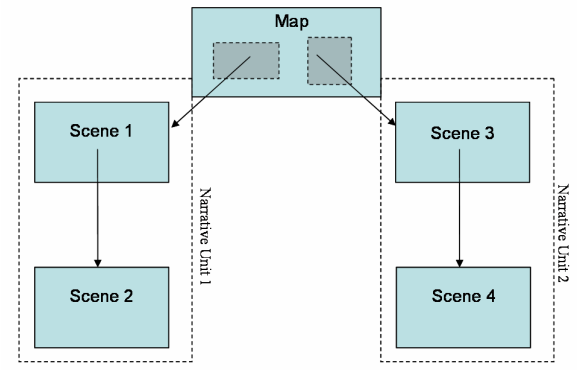

Figure 2: *Hypervideo example structure.*

Media used for case study examples are AVI videos with 352x288 pixels resolution and one Jpeg image with 140x90 pixels dimension. All the captions are created using XMT-A <Text> nodes.

4 Layout profiles

4.1 Profiles definition

Each layout profile defines a way to organize the audiovisual scene, establishing positioning, dimensions and graphical aspects (colour, texture, graphical widgets) of scene and of its *channels*. To demonstrate the use of layout profiles, we have defined two types of scene organization based on two aspects: presentation device type and graphical features. We have defined two layout profiles: the "Small device" layout and the "Web-based device" layout.

The "Small device" layout adapts the hypervideo presentation for portable devices like PDA.

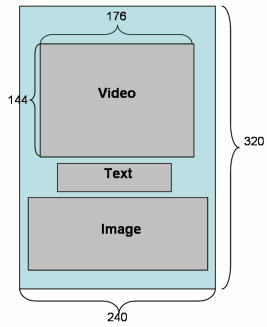

Figure 3*: "Small device" layout channels arrangement.*

Based on our choices, the scene *channels* are positioned as in figure 3. The presentation device is modelled for 240x320 dimensions, thus, there is lack of space for scene objects. Video *channel* placed on the top of scene has 50% of real video dimension. Image *channel*, is positioned on the bottom with 50% of original dimension. Text *channel* is positioned in the middle, with a 12-point font type.

The "Web-based device" layout target device is a desktop platform.

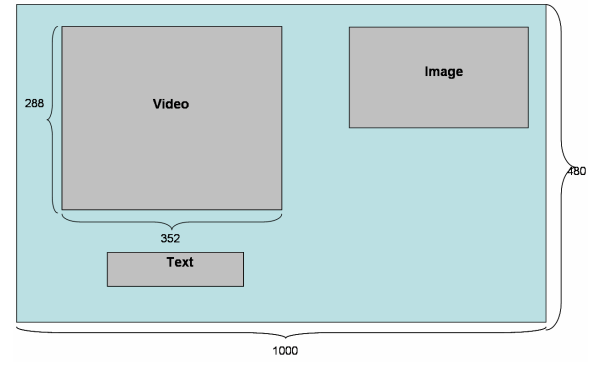

Figure 4: *"Web-based device" layout channels arrangement.*

The scene is modelled with 1000x480 scene size. As visible in the figure 4, it is organized as follows: the video *channel* is positioned in the top-left with its original size, the image *channel* is placed in the top-right also with original dimension and the text *channel* is positioned below the video with a 16-point font type.

4.2 Layout and media separation

Our working approach uses adaptation of hypervideo presentation layout through device profiles, without re-encoding of the MPEG-4 scene. Involved media are described in single XMT-A representations (one for each *narrative unit*) and encoded in single MP4 files separately from audiovisual scene containing layout information. Linking between scene layout and media is accomplished with an *inlining* technique. This technique consists in using <Inline> nodes, which allow to insert into the BIFS scene an external object referenced with an *Object Descriptor*, which in our work is an external BIFS stream (contained in an MP4 file).
XMT-A documents representing *narrative units*, are encoded separately. Each document contains an audiovisual scene of video dimension. Since every video scene has a spatio-temporal link opportunity, the audiovisual BIFS scene contains information related to the synthetic object used to represent the opportunity, and its position update information. Finally, the scene contains also, the *Object Descriptors* of each used media (in our case study examples two AVI video clips).

```
<XMT-A>
            …
<OrderedGroup DEF="Cha nnelOne" >
 <children>
  <Transform2D translation="-145 48" scale="0. 5 0.5">
   <children>
   <!-- Video narrative sequences -->
    <Inline DEF="Video" url="od:6"/>
    <MediaControl DEF="MC" url="" preRoll="true"/>
    <MediaSensor DEF="MS" url="od:6"/>
   </children>
  </Transform2D>
<children>
</OrderedGroup>
            …
<ObjectDescriptorUpdate>
 <!--OD linked media 1-->
 <OD>
  <ObjectDescriptor objectDescriptorID="od6"
binaryID="6">
    <URL URLstring="media/narrative1.MP4"/>
   </ObjectDescriptor>
 </OD>
</ObjectDescriptorUpdate>
            …
</XMT-A>
```

Channel Spatial Informations

External Scene Link

Media MPEG-4 Ext. Scene Object Descriptor

Figure 5: *Insertion of an external scene using XMT-A Inline node.*

The use of <Inline> nodes for linking media (*narrative units*) in audiovisual scene, allows to modify the graphical aspect of a hypervideo by simply changing the XMT-A of scene layout even at runtime. This technique allows adaptation of hypervideos presentations based on device characteristics.
Observing figure 5 is possible to understand how an audiovisual BIFS scene can contain another BIFS scene in hierarchical way.
The scene containing media is referenced in the main BIFS scene (the layout scene) as a stream of type *SceneDescription* stream and encapsulated in a new *Object Descriptor* (for each media used in the hypervideo). Subsequently, an <Inline> node inserts as a new *MO* the *Object Descriptor* containing the external scene. Now, it is allowed to use this *MO* modifying his spatial positioning; and his appearance, through XMT-A nodes like <Transform2D> (scaling, positioning) or <OrderedGroup> (grouping, layering). We have thus separating the compositions aspects of scene from media that, in form of *narrative units*, constitute the hypervideo storyboard.
In addition, in our example, when selecting a narrative unit from the interactive map, the <Inline> node is updated with a reference to the correct *Object Descriptor* containing the selected *narrative unit*. Thus, in the main XMT-A scene are included all the *Object Descriptors* referencing the external media in form of MP4 files.

A good example of application is the automatic adaptation of content when the presentation device changes. This application can be done, creating one layout XMT-A scene for each device and linking it to the same media (e.g. streamed from a server), maintaining on the device only one XMT-A file (encoded in MP4 format) with small space occupation (in order of Kbytes) advantage.
Dynamically structure-changing hypervideos can be supported using XMT-A <ServerCommand> nodes that allows a server-side scene update through a back-channel. In this case, layouts profiles can be more complex, containing *channels* update events at run-time.

4.3 Implementation problems

We have implemented our methodology examples using the GPAC framework to encode XMT-A documents into MP4 files and in order to playback the hypervideo through Osmo4 player; figures 6 and 7 shows case study examples implemented.

The use of hierarchical multiple MPEG-4 BIFS scene allows a real separation between layout and media data of hypervideo presentation. However, we have found some implementation problems due to MPEG-4/BIFS architecture. While BIFS allows connecting multiple scene description, in hierarchical manner, it also does not grant to pass events between the scenes. In fact, every inlined audiovisual scene creates a new nodes scope, thus it is not possible to pass events between parts of a scene that reside below different <Inline> nodes. In this case, a first scene is conveyed in the main scene description stream and a second part of the scene is inlined, accessed and pointed to (via URL) in the *Object Descriptor* stream of the first scene.

The lack of a communication channel between the two scenes scopes limits the possibilities of interaction. For

instance, it is not possible to call interaction events in the main scene that use *MOs* from the inlined scene. Another problem is the lack of synchronization between two streams used in different scenes (one in the main scene and one in the inlined scene). This problem comes because in the standard it is possible to use synchronization only between *Object Descriptors* and not between different streams.
We think that a good solution can be the definition of "events channels" that allow interaction between two BIFS scenes, saving the original behaviour of different scene scopes but allowing event passing through the scenes.

5 Conclusion

In this paper, we provide a possible approach of content adapting through different devices, in order to use layout profiles in MPEG-4 hypervideos presentations. We have described how to implement this approach in MPEG-4 XMT framework, using a hypervideo presentation example as a case study. We have denoted the power of this approach and the problems of implementing it.

Scaling to complex adaptation is possible but can suffer of communication problems between different scenes scope. We would like to propose to the MPEG-4 Group implementation of the "events channels" to increase scenes interactivity between different BIFS stream.
In the future, we hope to extend this adaptable approach with automatic layout profile generation based on use of devices semantic information.

6 Acknowledgements

We acknowledge GPAC project community for GPAC framework usage in our MPEG-4 hypervideo examples.

Figure 6: *Screenshot of "Web-based device" layout hypervideo example.*

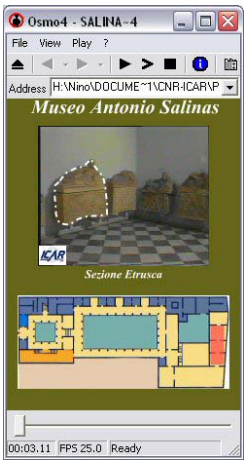

Figure 7: *Screenshot of "Small device" layout hypervideo example.*

References

[BRHO99] BULTERMAN D., RUTLEDGE L., HARDMAN L., VAN OSSENBRUGGEN J.: Supporting Adaptive and Adaptable Hypermedia Presentation Semantics. In *8th IFIP 2.6 Working Conference on Database Semantics: Semantic Issues in Multimedia Systems*, (Jan. 1999), Rotorua, New Zealand, (1999).

[Gag03] GAGGI O.: Synchronized Hypermedia Documents: a Model and its Applications. *Technical Report n. 05/2003*, Department of Computer Science, University Ca' Foscari of Venezia, Italy, (2003).

[GZ02] GRUNDY J., ZOU W.: An Architecture for Building Multi-device Thin-Client Web User Interfaces. *CAiSE 2002*, 728-732.

[Koe02] KOENEN R.: MPEG-4 Overview, ISO/IEC/JTC1/SC29/WG11/N4668, (Mar. 2002).

[LoB05] LO BUE, A.: Descrizione di Segmenti Video MPEG-7 e rappresentazione MPEG-4. *Bachelor Thesis*, University of Palermo, Italy, (2005).

[MPG02] Information Technology - Coding of audio-visual objects. Part-1: Systems, ISO/IEC/JTC1/SC29/ WG11/N4848, (Mar. 2002).

[MTLT05] MACHÌ A., TRIPICIANO M., LO BUE A., TRAPANI M.: Descrizione in MPEG-7 e rappresentazione in MPEG-4 di Video Objects per l'anchoring di hyperlink in video interattivi. *Technical Report RT-ICAR-PA-05-11* ICAR-CNR Dept. Palermo, Italy, (Oct. 2005).

[RVM02] ROMULUS G., VINCENT C., MATTHIJS D.: Optimizing hypervideo navigation using a Markov decision process approach. *ACM Multimedia 2002* (2002), pp 39-48.

[SBI96] SAWHNEY N., BALCOM D., SMITH I.: HyperCafe: Narrative and Aesthetic Properties of Hypervideo. In *ACM Proc. Hypertext 96* (1996), pp. 1-10.

[VLFO*01] VAN DER DONCKT J., LIMBOURG Q., FLORINS M., OGER F., MACQ B.: Synchronised, model-based design of multiple user interfaces. In *Proc. Workshop on Multiple User Interfaces over the Internet*, (2001).