

A TEMPLATE-GUIDED AUTHORIZING ENVIRONMENT TO PRODUCE MPEG-4 CONTENT FOR THE WEB

Souhila Boughoufalah, Marc Brelot, Frédéric Bouilhaguet, Jean-Claude Dufourd

ENST, Dept ComElec, 46, rue Barrault, 75013 Paris, France

E-mail: { souhila, brelot, bouilhaguet, dufourd }@enst.fr

ABSTRACT

Today, content creation is not reserved to professional authors anymore. Many net surfers wish to create their own content to share them with their tribe: family, friends... Many tools are available to create media contents for the Web as Macromedia Flash. Nevertheless, Flash is a professional authoring tool and it is much too complicated for a simple user. In our solution, we provide pre-created templates to the author and all he/she has to do is to customize these templates with his A/V sources, texts, colors ... In addition to the ease of authoring, the multimedia content is delivered in MPEG-4 format. In this paper, we will demonstrate how, for the applications we want to address, MPEG-4 offers new possibilities in comparison with other formats and we present our template-guided editing tool Harmonia, its interface and architecture.

Keywords: MPEG-4, Authoring tool

1 INTRODUCTION

For its many advantages developed in section 2, MPEG-4 is our choice for multimedia document production. But authoring MPEG-4 scenes is a complex task : direct BIFS authoring is unsuitable for non experts. Harmonia provides means to create complex MPEG-4 scenes in a very simple way. The target public for this tool is non-professional authors and the authoring is achieved through the use of templates. Templates are MPEG-4 scenes that are created by professionals and are customizable by non-professional users. The author is given a template library and for each template, easy-to-customize features are provided. Some helper tools come with Harmonia to make special processing on movies, like video shape encoding, or to create special objects, like slide-shows, and to synchronize scene events.

In this paper, we introduce our Harmonia tool. We present the elements of a scene, the user interface, the structure of building blocks and the tool architecture.

2 WHY MPEG-4 ?

We think that this standard is well-adapted for the new multimedia applications needs by providing the following features :

?? **Integration**

MPEG-4 integrates many media types : 2D, 3D, video, audio, synthetic visual and audio, video with non rectangular shape, scalable video and audio...

?? **Synchronization**

MPEG-4 integrates classical synchronization between audio and video as well as more advanced synchronization between a media stream and a scene description stream. This synchronization is respected even in bad network conditions. If there is a delay on the media arrival time, the synchronized BIFS element won't be rendered at the terminal until the corresponding media arrives.

MPEG-4 AMD2 also provides the Flextime model which introduces a strong synchronization paradigm. Using relationship constraints, media objects and graphical objects are linked to construct a time graph. This specification enables synchronization of multiple streams coming from various sources.

?? **Streaming**

On the web, the classical model used for multimedia presentations is that a scene description (for instance an HTML page or a VRML scene) is downloaded once, and then played locally. In the MPEG-4 model, a compressed scene description, BIFS [ISO 2000], is delivered over time. The initial scene is delivered in a special update and further changes of the scene are performed through updates. MPEG-4 integrates also a delivery framework DMIF which covers all useful delivery scenarios so our content can be delivered over the Web within http, multicast or RTP with minimal changes (if any). Because of these features and because it is a standard, we believe MPEG-4 will be successful even against the currently well-established proprietary formats.

3 HARMONIA : AN MPEG-4 TEMPLATE-GUIDED EDITING ENVIRONMENT

Harmonia is a simple graphical, "drag & drop" editor. An author can choose a predefined template and insert objects like movie, slideshow... and replace an object in this template. Objects can be configured by modifying their attributes (size, color, ...). Complex objects like slideshow need to be configured in separate editors to set all the slides with the transitions related to each slide. The author can play his/her content at any time with an integrated

player or publish the MP4 file with the default scenario for http context.

3.1 Elements Composing a Harmonia Scene

To understand the elements composing a Harmonia scene, we give a few definitions.

A **BIFS object** is an MPEG-4 scene element.

A **Harmonia object** is an object specific to our authoring environment including “java” objects for customization purposes. These objects have a meaning only in the Harmonia editing environment.

A **Building Block (BB)** is the association of BIFS object(s) and Harmonia object(s). The building blocks can be of different types : (still) image, video, audio, text, slideshow, button.

BBs have generic properties which can be on or off depending on the template context: movable, resizable, deletable and replaceable, which comes with a list of allowable replacement types.

BBs have variants. When a user creates an object by drag and drop, a default BB for the selected type is added. For example, the “default” image BB resizes the image to the BB’s size: there are variants of the image BB that do not resize the media, or clip it, or show scrollbars... This is specially relevant for buttons. The tool comes with a small set of variants, and new variants can be added in a plug-in fashion.

All BBs have some common properties related to mouse sensitivity. BB instances can be a hyperlink: either the link is to another scene, and then it is a one-way link ; or the link is to another “page” inside the same scene, with a “go

back” button, the other page being populated with an inlined scene. BB instances can also be draggable within the scene. BB instances can finally be the source of actions.

Harmonia provides the author with a notion of **behavior** which can be triggered by interaction or time. An action is defined by four parameters: source object, target object, type of event e.g. button click, time ... and requested behavior: e.g. start media, show object ...

A **Template** is an entity including several building blocks. Some templates are empty and the user can insert any building block at any location and set the needed behavior. Other ones are more specific because more complex. For example, the mosaic template is a program grid comparable to what most satellite operators offer to their customers. In this case, the author has only to customize the A/V content of the grid elements.

3.2 Structure of a Building Block

The persistent part of a BB is a set of a BIFS scene, a Harmonia script and a set of Java classes. Table 1 shows a sample with a BIFS tree, the script and a short description of what functionality each node helps creating. The Java code has two functions: (1) parsing the script and establishing the link with the in-memory representation of the BIFS ; (2) implement any BB-specific customization.

The BIFS structure is a generic canvas and depending on the Harmonia script, not all the options are open to the user. For example, on the 4th line of the script, the “true” means that the BB is movable.

BIFS Component of the BB	Description	Harmonia Script
<pre> Anchor [ID_55] ├── PlaneSensor2D [ID_56] ├── Transform2D [ID_1] │ ├── TouchSensor [ID_57] │ ├── Switch [ID_2] │ └── Transform2D │ ├── Shape │ │ ├── Rectangle [ID_3] │ │ ├── Appearance │ │ │ ├── Material2D [ID_4] │ │ │ └── LineProperties [ID_5] │ │ └── Shape │ │ ├── Bitmap │ │ └── Appearance │ │ └── ImageTexture [ID_6] │ └── Conditional [ID_7] └── Conditional [ID_8] </pre>	<p>Creates the hyperlink capability</p> <p>Creates the draggable capability</p> <p>Position of the BB (incl. dragging)</p> <p>Allows actions and internal links</p> <p>Creates the Show/Hide capability</p> <p>Background of the BB</p> <p>Color and transparency of bg</p> <p>Border of the bg</p> <p>Image</p> <p>Implements the Show action</p> <p>Implements the Hide action</p>	<pre> Image "Image" { top, url "ID_55" enabled "ID_56" translation "ID_1" true enabled "ID_57" size "ID_3" false emissiveColor ,transparency "ID_4" lineColor, width, lineStyle "ID_5" url "ID_6" EventIn { activate 223 "ID_7" reverseActivate 224 "ID_8" whichChoice 225 "ID_2" } EventOut { isActive 226 "ID_57" isOver 222 "ID_57" } } </pre>

Table 1: MPEG-4 scene and corresponding Harmonia Script for an Image Building Block

The last part of the Harmonia script, about EventIn and EventOut, defines the behavior of the BB. The EventIn spec concerns the actions that this BB can execute, in this case Show or Hide: the first line points to the Conditional implementing the Show, the second line points to the Conditional implementing the Hide and the last line points to the Switch node in order to set the initial visibility state. The EventOut spec concerns the events that this BB can generate: boolean events isActive and isOver from the TouchSensor.

Thanks to the generic architecture of Harmonia, building blocks of existing types or of new types can be easily added, allowing customization of Harmonia for specific application areas and catering for particular user group needs. For the specific BBs Slide-show and Movie, the customization is done in different helper tools that have their own internal formats. Specific APIs enable the communication between the different environments and Harmonia performs the translation from these formats into MPEG-4 format.

3.3 Harmonia Architecture

Harmonia is a layer on top of MPEG-PRO providing what was referred to as the Template Manager in [BOU1 2000] [BOU2 2000]. In the current version, templates are customized at a static time 0 and don't include any BIFS-

updates. Only the objects Movie and Slide-Show have an intrinsic time and can therefore be synchronized (through the combination of startTime field of Movie and TimeSensors nodes).

In Harmonia, the MPEG-PRO part is responsible for the management of an MPEG-4 scene from encoding/decoding – composition - rendering – to MP4 management. Harmonia is responsible for the management of a Harmonia scene and Harmonia objects with the customization information and the conversion into BIFS of some Harmonia objects once customized.

MPEG has defined an exchange file format named MP4 [ISO 2000]. MP4 files include the MPEG-4 elementary streams: BIFS, object descriptor and media. Harmonia uses MP4 files to store templates, building blocks and partially customized scenes. These MP4 files contain a standard atom ? 'skip', designed to encapsulate opaque information? to store all non-standard information. Customization information is stored in this atom, as well as building block information. In final scenes, all customization information has been removed, so final scenes are not editable any more within Harmonia.

The usage scenario for final scenes is "http pseudo-streaming". This consists in organizing the MP4 file in such a manner that the information needed to start playing is at the beginning of the file, and the media samples are

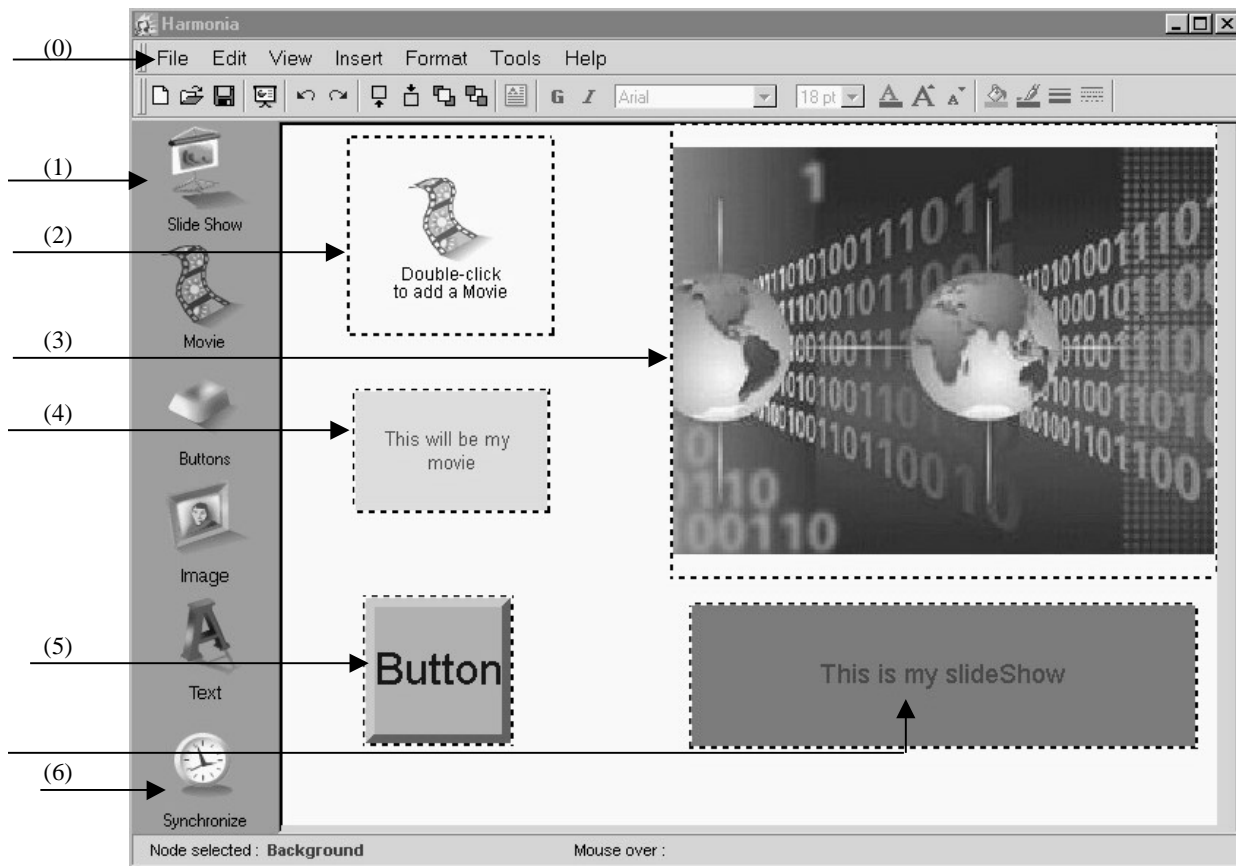


Figure 1 Harmonia User Interface

interleaved. Thus, the terminal can start playing the scene before the end of the downloading. There is no guarantee about quality of service.

3.4 User Interface

Launching Harmonia, the user obtains the screen shown on Figure 1. From the main menubar (0), he can open a template to customize it, or he has the ability to create content from scratch by opening an empty template and adding objects to it. Adding an object is done by dragging from the Object Bar (1) a slide-show, a movie, an image, a button or a text and drop it on the "view frame". The user may want to choose a variant of the chosen BB by invoking the contextual popup menu. The object needs also to be customized. Customizing the slide-show means opening a helper tool : the slide-show editor that enables to choose the slides, set the transitions between the slides, set the slides duration, etc... Customizing a movie means opening another helper tool: the movie editor allows video import, simple video editing and can invoke a semi-automatic segmentation tool to create a non-rectangular video.

On Figure 1, there is a customized slide-show (3) represented by the first slide and a movie object that has been added to the template (2) but not customized yet. The button (4) has been customized as a graphic button by setting the text of the button, the characters size, the background color: another choice is an image button. It is also possible to set the behavior of objects by enabling the triggering of some actions upon the some events. Events are: mouse over, mouse click, media start, media stop. The set of allowed events is easily extensible, so that other events like mouse down and up, or mouse enter and exit, or even events from other input devices can be taken into account. Actions are: show and hide, start and stop (for movies and slide shows only, of course), link to a URL. For example, when the event Mouse Click on the object Button occurs, it triggers the action "Play" for the slide-show. Actions are set via a property window attached to each object. Each object may be customized through a property dialog, with tabs for general properties, actions, and possibly BB-specific properties.

To give a temporal dimension to the presentation, it is possible for the author to launch the synchronizer (6). It is another helper tool that enables to synchronize objects that have an intrinsic time (starting with a startTime) : movie, slide-show, sound. Once all the content is customized, the author can view (View menu) the result through the integrated MPEG-4 player. He can also save the content in an MP4 file.

4 ACKNOWLEDGEMENTS

The work described here is the work of ENST within a project directed by TDK Corporation, and with the following partners : Telecom Italia-CSELT, Ecole des Mines de Paris, Optibase and Ness IT.

Harmonia is written in Java. The helper applications were implemented in other languages by Ness and Ecole des Mines de Paris. The integrated MPEG-4 player is implemented by Telecom Italia-CSELT.

5 CONCLUSION

In this paper, we have presented our solution for authoring MPEG-4 contents for a non-professional public. We have demonstrated why we think that MPEG-4 is the best format that fits our needs for our contents. We have presented the template-guided interface and explained the architecture that extends the MPEG-PRO tool.

Our future work will focus on :

- providing libraries of templates and building-blocks addressing different applications and for different markets ;
- providing new ways of editing synchronization constraints ;
- allowing BIFS updates in templates ;
- switching to an XMT-based representation of building blocks to integrate the BIFS and the script.

6 REFERENCES

- [ISO 1999] ISO/IEC 14496-1:1999, MPEG-4 Systems specification.
- [ISO 2000] ISO/IEC 14496-1:1999/AMD1, Amendment 1 to MPEG-4 Systems specification, including MP4 file format.
- [BOU1 2000] "MPEG-PRO an MPEG-4 Authoring System" Souhila BOUGHOUFALAH, Jean-Claude DUFOURD, Frédéric BOUILHAGUET, ISCAS 2000 IEEE International Symposium on Circuits and Systems May 28 - May 31, 2000 GENEVA.
- [BOU2 2000] "MPEG-Pro, an Authoring System for MPEG-4 with Temporal Constraints and Template Guided Editing", Souhila BOUGHOUFALAH, Jean-Claude DUFOURD, Frédéric BOUILHAGUET, ICME 2000 IEEE International Conference on Multimedia and Expo July 30 - August 2, 2000 NEW YORK