

A Template Based Production Chain

[Using richmedia templates to assist users during content production]

Renaud Cazoulat^{*}
France Telecom R&D
Rue du Clos Courtel, 4
35512 Cesson-Sévigné
+33299124287

renaud.cazoulat@orange-ftgroup.com

Thierry Le Bris[†]
France Telecom R&D
Rue du Clos Courtel, 4
35512 Cesson-Sévigné
+33299124332

thierry.lebris@orange-ftgroup.com

ABSTRACT

Mobile terminals such as smartphones and PDA are able to produce picture, audio and video contents. However, the creation and publication of a mixed content document remains quite difficult and even does not exist on mobile devices. The authoring motivates our work.

In this paper we will describe a framework based on rich-media templates to assist users during all stages of a complex content production from expert user to end user. By complex content we mean mixing videos, texts, audio, pictures in a time-dependant document¹[4]. The richmedia language we have selected is MPEG-4 BIFS (BInary Format for Scenes)[3] for its high level integration of many media type and streaming capabilities which is decisive in a mobility context.

1. INTRODUCTION

Richmedia languages such as SMILE, SVG, MPEG-4 BIFS or Flash have been designed to enhance HTML possibilities in terms of displaying and user interaction. The MPEG-4 BIFS description format is well-adapted for tiny application or rich publication needs (specialy integration of different media type, synchronization and streaming) and fits with the *template* concept[1]. However, the authoring of a rich-media publication is still uneasy and its deployment generate classical hosting and compatibility problems[5].

We present a *work in progress* framework based on MPEG-4 BIFS templates that purpose three using level: richmedia programmer, project designer and end user who only had to fill a wizard.

This work is a continuation of an alike ENST web-targeted

^{*}Research engineer at FT R&D

[†]Rennes I University Phd. student, comanged by R. Cazoulat and K. Bouatouch

¹Also called "richmedia" content

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Mobimedia'07, Month 8, 2007, Nafpaktos, Aitolokarnania, Greece
Copyright 2007 ICST 978-963-06-2670-5 .

work[1][2]. On PC, proprietary formats such as Macromedia-Flash are well-established, but on mobile we think MPEG-4 BIFS is a good technology. On mobile side we use an internal MPEG-4 BIFS player developed in J2ME[6].

This system will be suggested in the European project CitizenMedia.

2. A FRAMEWORK BASED ON TEMPLATES

A template is a simple scene described in the targeted language, as said before we use MPEG-4 BIFS. It represents a generic functionality inside an application, for example a slideshow, a menu, a simple play/pause video player. The MPEG-4 BIFS standard includes a node called *Inline* which loads the MPEG-4 BIFS file set in parameter upon the displayed scene. With this node we can build an inclusive graph considering each file as a template.

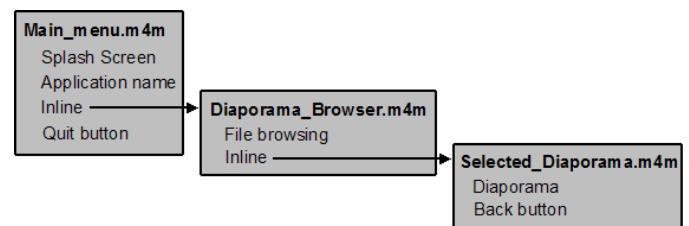


Figure 1: Template inclusion with Inline node

In this example we have a main menu with its properties linked to a file browser which display a list of all available diaporama (in local mobile memory or on a server, it's a property of this template). This file browser is able to launch a diaporama. It's a simple application with three templates, we consider that diaporamas already exist.

2.1 A template

Each template is defined by : a graphic representation (except in case of non-graphical feature like a background sound module), a set of properties and a list of other templates it can be linked with.

The following diagram shows a template for a graphical menu which is defined by a title and four items. This template will be linked to four others templates and can be the starting point of a richmedia application or can be just a link into the application state-graph.

Our implementation of this template in MPEG-4 BIFS consists in a graphical menu linked to three others scenes, which names, link-names and physical addresses of links can be changed. It can be left empty for a future GUI wizard. (active template).

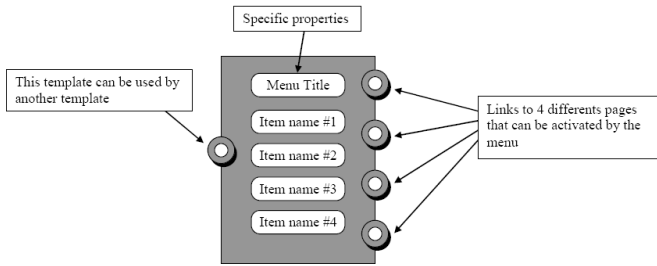


Figure 2: Graphical representation of a menu template

2.2 Linking

A richmedia application can be designed by a set of templates linked all together. For example the video browsing application can be represented by the following diagram. Each template represents a functionality.

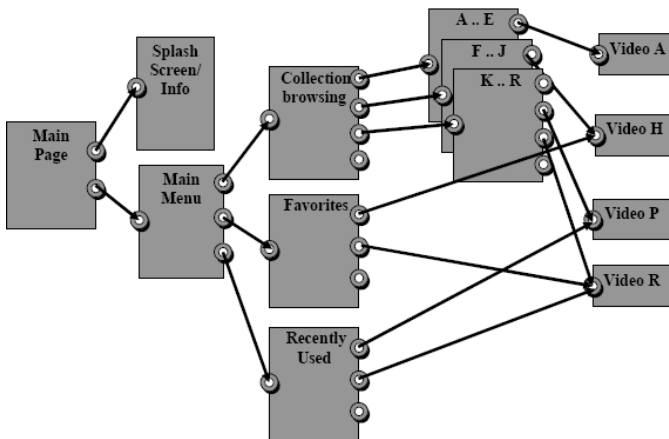


Figure 3: Richmedia application project

2.3 Inside a template

A template is made of two information pieces.

- The first one is a textual definition of a generic scene improving the targeted feature (e.g. the MPEG-4 BIFS file).

```
PROTO CubePhoto [
#DEF PATHTOPICTURES
  exposedField MFString url [""]
#DEF TITLE
  exposedField SFString title ""
]{
Group
{
```

```
children
...
```

- The second information is a description of the possible customization of this template (e.g. "What are the informations that are missing ?"). It's a way to indicate that there are holes in the template and for each what kind of information is missing and how to put it in the final product (e.g a XML description file).

For that purpose the description has two different sections describing the template properties and the instantiation process using the following XML syntax.

```
<area name="Title" type="text" >
  <param name="co_text" type="string" default="">
    My title
  </param>
</area>
```

The building instructions use the content of each property by referencing their name and consists basically in a sequence of *pattern/new_value* pair. When the pattern is recognized in the scene definition its value is replaced by a new one.

```
<set field="TITLE.title">"$(co_text)"</set>
```

This new value is any combination of char and property values using the simple syntax \$(id).

3. THE CONTENT CREATION PROCESS

There are three major actors in the process of scene creation.

- The expert that designs the templates.
- The advanced user that creates projects by assembling templates together and starting filling holes.
- The end-user that choose a project and finish filling missing information in order to customize the richmedia application.

The main benefit of this architecture is that each actor can work independently with specific tools adapted to his level of technological knowledge. The expert may work with complex tools and concepts to create scripts or adaptative graphical objects. He works directly with the targeted technologies (MPEG-4 BIFS but can be any structured richmedia description language such as SMILE, SVG). The advanced user will use a graphical tool to assemble templates, fill some properties in order to build a project. The graphical tool can take advantage of propertie's description to automatically deduce a GUI from a project: a wizard. This GUI is the only thing the end user will see.

The feature is really important as the most critical creation time will be spent during project creation. Further, as any kind of templates can be created, building an application will be like playing with blocks with simple functionality.

Finally the project definition is used jointly with the properties definition to produce a wizard that enables end users to fill missing information without having to deal with the complexity of the project.

4. CONCLUSION AND NEXT STEP

This template system has been successfully used to create some richmedia scenes with a high level of interactivity. A first version of a graphical tool has been designed in our lab to test the ability for advanced users to design complex projects used by web based wizard and will be proposed to the IST project CitizenMedia.

The next step is to introduce the concept of *active templates* where advanced scripting is included in order to have content created from projects that will be able to use automatically existing contents (e.g. XML Data resources on the web), making a good base for advanced co-creation of richmedia content linked on network resource.

5. REFERENCES

- [1] S. Boughoufalah and M. BreLOT. A template-guided authoring environment to produce mpeg-4 content for the web. *ENST Online Document*.
- [2] S. Boughoufalah and J. Dufourd. Mpeg-pro and mpeg-4 authoring system. *ISCAS 2000 IEEE International Symposium, GENEVA*, May 2000.
- [3] M. P. E. Group. Binary format for scene description: Bifs. <http://www.chiariglione.org/mpeg/standards/mpeg-4/>.
- [4] T. Lemlouma and N. Layaïda. Content interaction and formatting for mobile devices. *ACM Symposium on Document Engineering*, November 2005.
- [5] C. Roisin. Outils pour la production de presentations hypermedias. *INRIA summer school 2004*, September 2004.
- [6] I. Sun Microsystems. Java platform, micro edition (java me). <http://java.sun.com/javame/>.

APPENDIX

A. SCREENSHOTS



Figure 4: A project using templates in our GUI project builder: A simple menu with link to two cube slideshow



Figure 5: A web interface to fill a cube-diaporama wizard



Figure 6: Result in MPEG-4 BIFS for the diaporama wizard (cube effect)