Playback of Mixed Multimedia Document

Cyril Concolato   Jean Le Feuvre

Multimedia Group, Department of Image and Signal Processing
TELECOM ParisTech, LTCI UMR 5141
46 rue Barrault, 75013 Paris, France
{concolato, lefeuvre}@telecom-paristech.fr

ABSTRACT

Many multimedia languages exist today to describe animated, interactive, 2D or 3D graphics and media elements, and each language has its merits. We studied the problems underlying the integration of all these languages in a single player. We present here the result of this work, and in particular, we demonstrate the mixed playback of SVG, BIFS, LASeR, Flash or VRML/X3D content.

Categories and Subject Descriptors

I.7.2 [**Document and Text Processing**]: Document Preparation – *Languages and systems, Multi/mixed media;* H.5.1 [**Information Interfaces and Presentation**]: Multimedia Information Systems.

General Terms

Design, Experimentation, Languages, Standardization.

Keywords

BIFS, Mixed Documents, Multimedia Player, SVG, VRML.

1. INTRODUCTION

Multimedia documents are now composed of multiple media elements (audio, video, graphics, images or text), described using multiple formats. Among these formats we can cite:

- File formats relying on XML-based languages such as XHTML, SVG or SMIL which integrate well with Web technologies like CSS or DOM;
- Other textual formats such as X3D or VRML which are focused on describing 3D graphics and 3D worlds.
- Streaming formats (binary or textual) developed for broadcasting applications such as Flash, MPEG-4 BIFS or MPEG-4 LASeR.

Playback of such documents implies decoding or parsing the file or stream; composing the presentation (applying styles, performing animations, handling user events and synchronizing audio and video streams); and finally rendering the visual and aural parts of the presentation. Performing all these steps in an efficient manner is a well-known problem which has been addressed in both research and commercial products. There are many browsers available for XHTML; many players for SVG, SMIL, BIFS or VRML and the widely distributed Flash player.

Each language having its benefits, it would be interesting to mix several languages in the same mixed document. A mixed, also called compound, document is either a collection of different languages in the same file (e.g. using XML namespaces); or a collection of files expressed in different languages and linked together, when the syntactical structures cannot be packaged together (e.g. textual files and binary streams). This has been the purpose of some research activities like [1] and of the W3C Compound Document Formats Working Group. However, both activities currently either only envisage mixing Web languages (XHTML and SVG, or SVG and MathML…) or integration using embedded viewers. While this is not an easy task, mixing content from very different languages like BIFS and Flash or SVG and X3D in the same presentation is also a hard task. This implies handling different synchronization models (stream based vs. document based); different animation models (key frame vs. interpolation); different event models; or different rendering models in particular between 2D and 3D languages.

In this paper, we describe the result of the implementation of an open-source player [2] capable of playing mixed 2D/3D content from the following languages SVG, BIFS, LASeR, Flash, VRML/X3D. The rest of this paper is organized as follows. Section 2 describes the important design choices which enable mixed playback. Section 3 describes demonstrations that will be presented. Finally, Section 4 discusses future work in this area.

2. DESIGN OF A MIXED FORMAT DOCUMENT PLAYER

In this section, we present the main issues that we solved in the design of a mixed multimedia document player. A simplified architecture for this player is depicted in Figure 1. It illustrates the different areas where issues had to be addressed.

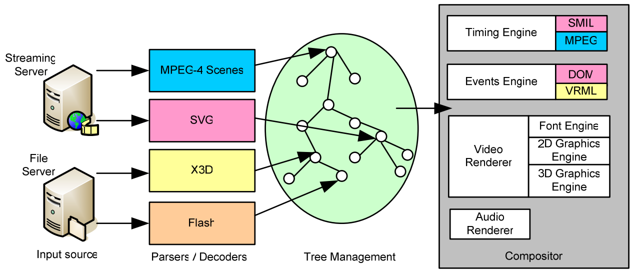

Figure 1 – Simplified Architecture of the GPAC Mixed Multimedia Document Player

2.1 Scene Tree Management

The first task when playing mixed languages presentation is to distinguish elements from the different languages and to create appropriate structures. In our implementation, nodes are decoded and parsed, according to their native language and a base node

Copyright is held by the author/owner(s).
DocEng'08, September 16–19, 2008, São Paulo, Brazil.
ACM 978-1-60558-081-4/08/09.

structure is created. However, the structuring of the different languages can be very different: for example the BIFS language contains many types of nodes, each node described by few attributes (~4); whereas the SVG language contains fewer nodes (~50) possibly with many attributes. Therefore, our implementation uses a generic structure for DOM nodes and specialized typed structures for other languages.

2.2 Media handling

Almost all multimedia languages define the ability to embed content like images, audio or video streams. They even allow embedding of other multimedia languages. This is for example the purpose of the SVG *animation* or *foreignObject* elements or of the BIFS or VRML *Inline* or *AnimationStream* elements. However, designing a common loading algorithm is a problem. In our implementation, we unified loading algorithms for streams and file by considering progressive loading of XML document as special case of stream, and in both cases enforce an exclusive access to the scene tree between the loader and the compositor. We additionally added the notion of dictionary of resources which avoids, in some languages, multiple loading of the same resource.

2.3 Graphics and Text handling

Our experience shows that even though the expressivity of existing languages in the graphics and text is different (filters in SVG, texturing BIFS and Flash, different text layout and decorations), there is no fundamental difference in the graphics and text representations. Our implementation uses the same underlying graphics and text engines for all languages. The only additions are the detection of different coordinate systems (top-left vs. centered origin, y-axis orientation) and the application of a language intermediary transformation matrix. But, an important problem was the mixing of 2D and 3D primitives in the same document. For that purpose, we introduced the use of cached 3D or 2D layers and modified the picking engine which uses different methods for 3D or 2D objects (ray-tracing or not).

2.4 Timing, Animation and Synchronization

Concerning timing, we had to handle two actions: finding a common algorithm to handle intra-language synchronization (including multiple timelines) and defining a method to handle inter-language synchronization. For the first action, since most advanced languages support multiple timelines, have similar notions of animations and of media synchronization, the common algorithm was simple. However, insuring synchronization of elements from different languages in the same presentation was an issue. To solve this problem, our implementation groups the handling of all timed elements of all languages in a single pass, while the all timelines are frozen.

2.5 Interactivity

All modern languages share the possibility to handle user events such as key pressed, mouse move or clicks, using the concepts of event listener, event propagation and scene modifications. However, VRML/X3D or BIFS use an explicit event propagation using Route, while DOM-based languages use capture, bubbling and target phases. Additionally, we added a mechanism to detect the type of event to use, based on the target node namespace, and added a mechanism for event forwarding across documents. Finally, all modern languages relying on scripting for advanced interactivity, we validated AJAX concept in all these languages.

3. DEMONSTRATIONS

We present here three demos:

- BIFS + SVG or SVG + BIFS: This first demonstration shows animated and interactive graphical content mixing BIFS and SVG. Its purpose is to demonstrate that a player for OMA RME, 3GPP DIMS or T-DMB with a common core is possible.
- SVG + Flash: Cohabitation of Flash and SVG appears with the convergence of Web and Mobile TV application. This second demonstration shows how Flash media elements can be used in scenes which feature SMIL animation elements or W3C-like interactivity.
- SVG + X3D or X3D + SVG: Despite existing 3D standards and available browsers, 3D content is not very widespread on the Internet. However, many simple 3D features could be used in 2D content to enrich the quality of multimedia presentations. The last demonstration, illustrated in Figure 2, shows mixing SVG and X3D content, with either language as root language.

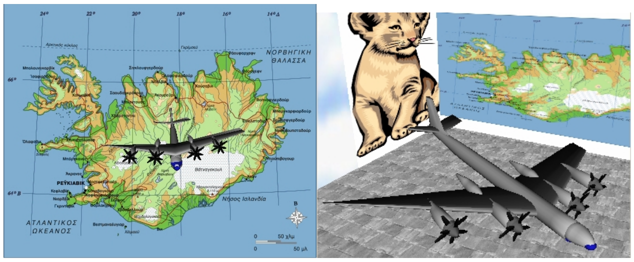

Figure 2 - Mixed SVG and X3D presentations

4. DISCUSSIONS AND FUTURE WORK

The purpose of this work was initially a proof of concept that all languages are based on common functionalities and that an integrated player was possible with little overhead. The opening of the source code of the player follows the philosophy of sharing these results. In the future, we want to investigate cross-language technologies like W3C XBL or Styling and to work on some of the current limitations such as event and scripting communications between documents.

5. ACKNOWLEDGMENTS

This work was partially supported by the European Commission under the INTERMEDIA Network of Excellence (038419).

6. REFERENCES

[1] Vuorimaa, P., Ropponen, T., von Knorring, N., and Honkala, M. 2002. A Java based XML browser for consumer devices. In *Proceedings of the 2002 ACM Symposium on Applied Computing* (Madrid, Spain, March 11 - 14, 2002). SAC '02.

[2] Le Feuvre, J., Concolato, C., and Moissinac, J. 2007. GPAC: open source multimedia framework. In Proceedings of the 15th international Conference on Multimedia (Augsburg, Germany, September 25 - 29, 2007). MULTIMEDIA '07.