MPEG-4 Systems

Alexandros Eleftheriadis[*a], Danny Hong[†a], and Hari Kalva[‡b]

[a]Dept. of Electrical Engineering, Columbia University, New York, NY, USA
[b]Flavor Software, Inc., New York, NY, USA

ABSTRACT

We briefly describe the process for creating an MP4 file and introduce the software tools used for the creation. Then, we describe the architecture of an MP4 player – Flavor Player – that implements the MPEG-4 Systems specification. The Flavor Player implements 2-D composition and depth ordering of objects, object animation, user interaction, MPEG-J, IPMP framework, and MP4 file support. Additionally, we describe a simplified version of the Flavor Player – Mild Flavor – that only implements the Object Descriptor Profile. Unlike the Flavor Player, Mild Flavor is also used to create and edit MP4 files in addition to playback.

Keywords: MPEG-4 Systems, MP4, MPEG, Flavor, Flavor Player, Mild Flavor

1. INTRODUCTION

MPEG-4 addresses the "generic coding of audio-visual objects." In contrast to all other existing audio or video representation standards, MPEG-4 adopts an object-based approach for content description: the content is assumed to be constructed out of individual and independent entities called objects, which are separately encoded. These objects include, for example, arbitrarily shaped natural video, graphics, natural or synthetic audio, face or body animation etc. MPEG-4 has defined a number of representation tools to address the coding needs of a large variety of media. These tools extend much beyond the ones used in MPEG-1 and MPEG-2, which only addressed natural video and audio at combined bit rates of 1-20 Mbps. For example, MPEG-4 includes mesh coding tools, scalable coding of still images using zero-tree wavelets, face animation parameter coding, etc.

Coding of such objects, however, is only the first step into constructing a complete multimedia scene. Additional information is needed in order to: 1) describe how these objects should be placed in space and time, 2) how they may interact with each other and the end-user, 3) how to multiplex all this information into one or more streams for delivery over a variety of networks, and 4) ensure proper synchronization among the various streams. This information is the realm of the MPEG-4 Systems specification (Part 1 of the MPEG-4 standard) [1], which is also responsible for the overall architectural definition of MPEG-4.

In MPEG-4, content representation is separated into three major entities: object descriptors, scene description, and coded audio-visual data. A fourth category, object content information, can optionally be used as well [2]. In the following section, we describe the process for generating the three major entities and combining them into an MP4 file. Then, two different MP4 players (Flavor Player and Mild Flavor) and their architectures are described.

2. FLAVOR PLAYER

The Flavor Player is an application that plays back MP4 files. It implements a large subset of the MPEG-4 Systems specification: Complete 2-D Graphics Profile, Complete 2-D Scene Graph Profile, Main MPEG-J Profile and Object Descriptor Profile. In the following, we describe how to create an MP4 file that uses the tools included in the profiles. The profiles are defined in [1].

2.1 Content Creation

There are two main steps involved in creating an MP4 file. First, each audio-visual object is converted into an elementary stream that contains a series of *access units*. An access unit is the smallest individually accessible unit of data in an elementary stream to which timing information can be attributed. For example, in a video object, each access unit

[*] eleft@ee.columbia.edu
[†] danny@ee.columbia.edu
[‡] hari@flavorsoftware.com

Multimedia Systems and Applications V, Andrew G. Tescher, Bhaskaran Vasudev, V. Michael Bove, Jr., Ajay Divakaran, Editors, Proceedings of SPIE Vol. 4861 (2002) © 2002 SPIE · 0277-786X/02/$15.00

corresponds to a frame. Each access unit is encapsulated into a structure that contains timing and random access information, called the *Sync Layer*. We have created a software tool, MPEG-4 SL Packetizer (see Figure 1), that packetizes a given object into a series of access units along with extra information that indicates the format of the data and the configuration information required by the decoder. The extra information is later used in generating a corresponding object descriptor.

Once the objects that are needed during a presentation have been created and converted into corresponding elementary streams (ES files), a textual file containing the scene description is created. The scene description information describes how the various objects are positioned in space and time, and also defines dynamic behavior and user interaction. The scene description refers to the elementary streams by means of the object descriptor identifiers. This completely decouples the scene description from the specifics of the encoding of particular objects. For example, in Figure 1, the MPEG-4 Video object can be replaced with an MPEG-2 Video object without any need of modification of the scene description itself.

Finally, all the ES files and the encoded scene description file are packaged into an MP4 file using our software tool, Donna. During the packaging process, Donna generates object descriptors in the form of an object descriptor stream as well as a special object descriptor called the *initial object descriptor*. The initial object descriptor contains two elementary stream descriptors for the object descriptor and scene description streams. Additionally, if an object is to be protected, then the MPEG-4 SL Packetizer interacts with a corresponding IPMP System and generates IPMP descriptor and/or IPMP stream. Figure 1 shows the MP4 file creation process.

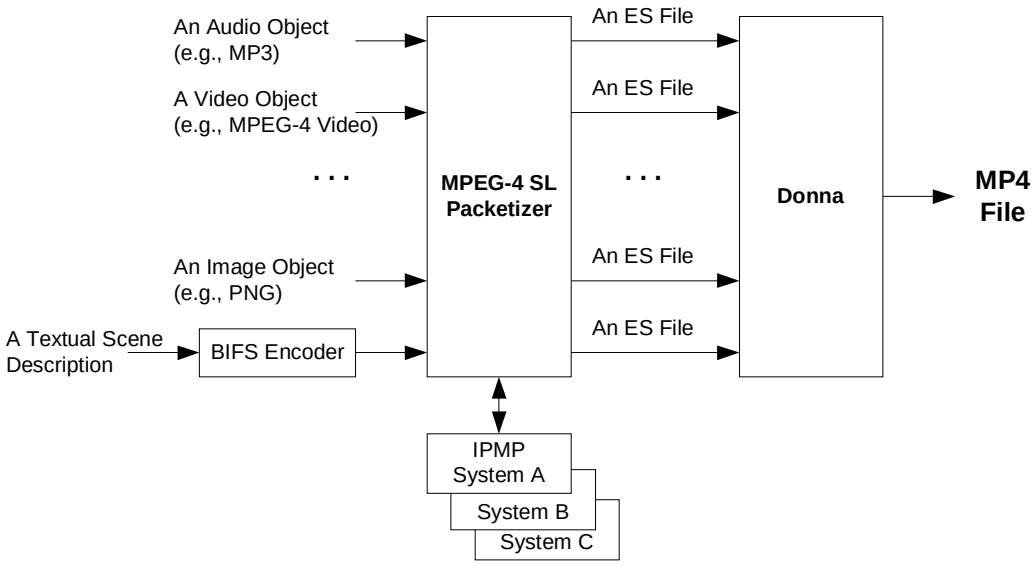

Figure 1: An MP4 File Creation Process

Java is also used to define user interaction behaviors. The resulting Java class is also packetized by the MPEG-4 SL Packetizer and passed to Donna along with other ES files.

2.2 Content Playback

Figure 2 illustrates the architecture of the Flavor Player. The player is comprised of five components (Controller, Data Reader, Data Sink, Compositor, and Renderer) and a set of decoders for the elementary streams (e.g., MP3 decoder and MPEG-4 Video decoder). The Controller is the core of the player and controls all other components. The Data Reader provides the DMIF Application Interface. The Data Sink contains a decoding buffer for each elementary stream and access units are pushed to the Data Sink by the Data Reader in a timely manner. The Compositor composes a presentation scene in accordance with the scene description and the Renderer is responsible for displaying the scene and capturing user actions. The player also includes an object descriptor (OD) decoder, a BIFS decoder, and a set of audio-visual object decoders. The current player includes an MP3 decoder, an MPEG-4 Video (Simple Profile) decoder and a PNG decoder. Below is the list of steps the player takes in response to an MP4 file playback command:

1. The player initializes the Renderer and the Controller.

2. The Controller initializes the Data Reader, and through the Data Reader, gets the initial object descriptor from the MP4 file. From the initial object descriptor, two elementary stream descriptors for the object descriptor stream and the BIFS stream are obtained.
3. For each one of the two streams, The Controller instantiates a corresponding decoder and makes sure that correct elementary stream is sent to the decoder through the correct buffer in the Data Sink. The connection from the Data Sink to the decoder is made by the Data Reader.
4. The Controller creates the Compositor.
5. For each elementary stream indicated by the object descriptors in the object descriptor stream, the Controller initializes a corresponding decoder and requests the Data Reader to open a channel and create a connection between the Data Sink and the decoder.

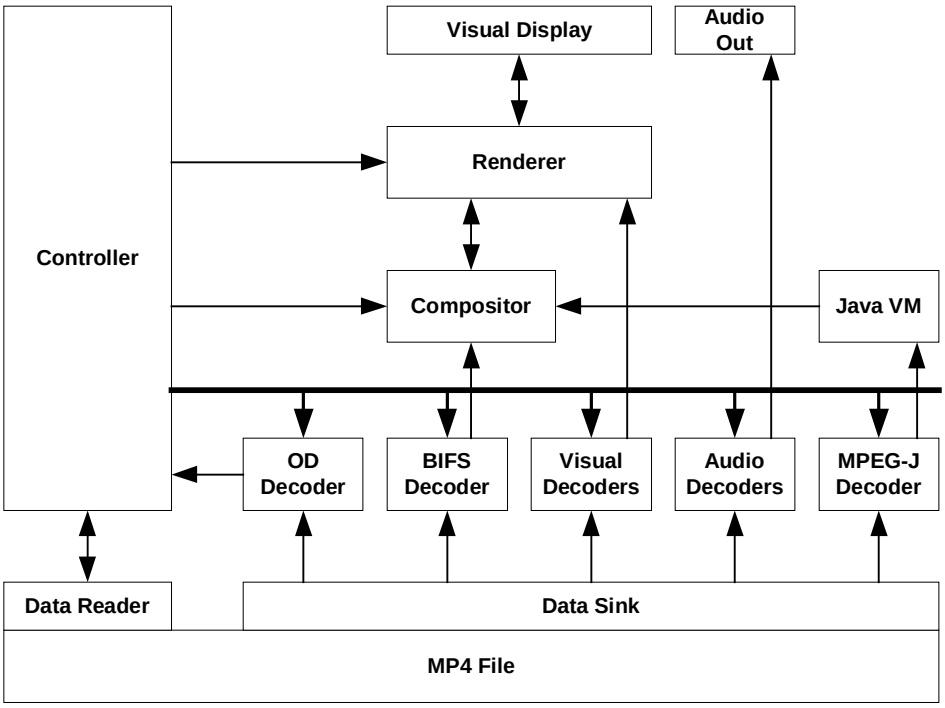

Figure 2: The Flavor Player Architecture

After the decoders have been instantiated and correct paths have been made between the elementary streams and the decoders, the access units are decoded in accordance to their corresponding timing information. In Figure 2, the regular arrows indicate data and control flow and the think arrows indicate a common interface, using which the Controller manages all the decoders. For a protected object, an IPMP descriptor must be present in order to instantiate a corresponding IPMP system. If the user/player has the rights to consume the object, the IPMP system makes the object available to the player (e.g., if the object is encrypted, it is decrypted prior to decoding).

2.3 Flavor Plug-in

Perhaps the most popular media player is Winamp by Nullsoft [3]. Taking advantage of its worldwide usage, we made a plug-in to the Winamp player so that it can also play MP4 files. The plug-in is a DLL version of the Flavor Player, with the same features that are available in the player. If an MP4 file contains visual objects, then a separate screen is popped up for the display.

Additionally, the Flavor Plug-in also gives a taste of a new patent-pending technology, ContentGUI, which allows each individual MP4 file to provide its own custom GUI to the MPEG-4 player. The concept takes the idea of player "skins" to the next level, by making the "look and feel" of the player part of the content. In fact, the same content rendering engine can be used to render the GUI itself, thereby making the full arsenal of content tools available for GUI design and operation.

This allows content providers and content creators to fully customize the user experience and preserve their visual, artistic identity. It is also essential for branding the player by major content outlets (e.g., Yahoo! or AOL).

3. MILD FLAVOR

Mild Flavor is also an MPEG-4 Systems player but with fewer tools implemented than the Flavor Player. This player only implements the Object Descriptor Profile and lacks the scene description information. Thus, the associated MPEG-4 content is less complex, and in addition to the playback of the content, the same player is also used to create and edit the content.

Figure 3: Mild Flavor

The object descriptors announce to the player different types of objects that are available in the MP4 file and also provide all the configuration information required for their decoding. Using the Object Descriptor Tool, Mild Flavor uses the MP4 file as a container for packaging different audio-visual objects into one file. The Object Content Information (OCI) Tool is also extensively used in order to provide ancillary identification information associated with each object in the MP4 file. Such information includes keywords in any language, ratings, author information, creation date, and so on.

Figure 3 displays an instance of the Mild Flavor player. The player can be viewed as an electronic "multimedia album," similar in concept to the electronic photo album. In addition to pictures and images, an MP4 file can also contain and organize other types of audio-visual objects (e.g., music and video). The left frame in the GUI of the player lists all the objects in the MP4 file, and selecting an audio or video object plays back the object and corresponding OCI is displayed in the two right frames. Selecting an image displays the image with its OCI. The display can be customized using an HTML template file.

The Mild Flavor player uses Windows Media Player Control and Web Browser Control to decode and render audio-visual objects and their OCI. In Figure 4, a new component, called View Generation, is defined and it is used to generate views for each object. A view is an HTML description of OCI that is displayed when corresponding object is selected. The view of each object is displayed using the Web Browser Control.

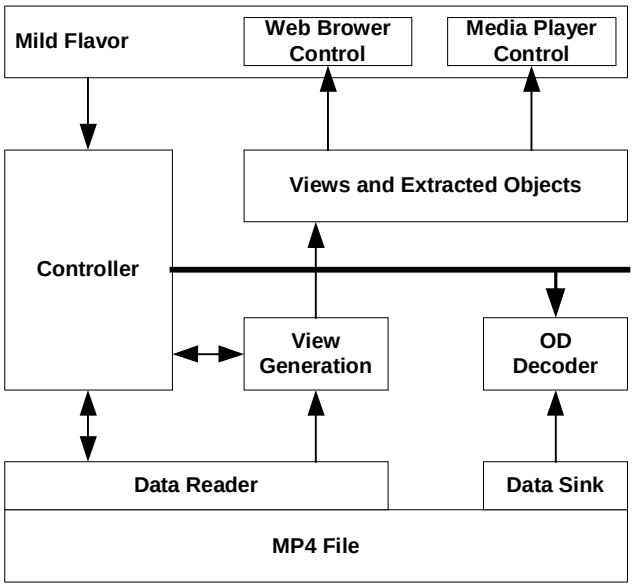

Figure 4: The Mild Flavor Architecture

4. CONCLUDING REMARKS

MPEG-4 Systems offers a framework that integrates audio-visual components (including audio, video, text, graphics, and 3D) in a seamless manner, enabling the next generation of interactive audio-visual services. The two applications introduced in this paper demonstrate compelling user experience that is not available anywhere else. The entire package of audio-visual objects combined with dynamic scene description and user interaction is delivered to users as a complete content experience in the form of an MP4 file.

5. ACKNOWLEDGEMENTS

The authors would like to acknowledge the members of Flavor Software, Inc. who built the applications and software tools mentioned in this paper.

6. REFERENCES

1. ISO/IEC 14496-1 International Standard, Information Technology – Coding Of Audio-Visual Objects – Part 1: Systems, 2001.
2. A. Eleftheriadis, "MPEG-4 Systems Systems," *Proceedings, SPIE Int'l Symposium on Voice, Video, and Data Communications*, Boston, MA, November 2000, Vol. 4209, pp. 1-12 (invited paper).
3. Winamp, http://www.winamp.com.