

WAM 9.1

On the Design and Implementation of an MPEG-4 Scene Editor

Deng-Rung Liu, Meng-Jyi Shieh, Yu-Chung Lee^{1,2}, and Wen-Chin Chen

¹Communication and Multimedia Lab.,
Department of Computer Science and Information Engineering,
National Taiwan University, Taipei, Taiwan

²CyberLink Inc., Taiwan

E-Mail: {drliu, feitian, cotta, wcchen}@cmlab.csie.ntu.edu.tw

Abstract

A prototype of MPEG-4 scene editor has been developed in the joint research of our laboratories and CyberLink Inc. With this scene editor, users can compose and edit object-oriented multimedia contents conforming to MPEG-4 [1] specification. The system is currently developed on Microsoft Windows platform. We describe in this paper the system architecture as well as the design and implementation issues of the scene editor. We believe our development experience can benefit future developers of MPEG-4 system.

Introduction

MPEG-4 is the state-of-the-art standard for describing the presentation and composition of multimedia contents. The contents integrate different media data in an object-based way. The specification of MPEG-4 includes media data access framework, video/audio codec, synthetic object presentation and integration method of different media data. We have developed a multimedia MPEG-4 scene editor based on the specification. The prototype system is developed on Microsoft Windows platform.

Architecture

Our system consists of five modules. They are *Editor Kernel*, *Render*, *SmartProxy*, *DMIF*, and *Codec Bank*. *Editor Kernel* is responsible for processing the construction of a scene description and activating the other modules in the initial phase of the program execution. *Render* module is responsible for composing different media objects and performing certain synchronization checks in rendering. The purpose of the *DMIF* module is to provide a uniform interface to access the data transmitted by different network hardware devices and protocols. Similarly, as there are different

codec standards for various media data, the *Codec Bank* concept (and thus module) is introduced to make the system codec independent. The above four modules have to communicate each other, such as exchange decoding result and map *DMIF* channel to physical URL of media object, to name just a few. Thus, the *SmartProxy* module is proposed to integrate and coordinate these modules.

Note that all the media data are parsed and processed in *Codec Bank* [3,4] module only. The *Codec Bank* module is currently implemented using the Microsoft DirectShow architecture. With the synchronization mechanism of the filter graphs of DirectShow [6], we can insert easily our proprietary synchronization mechanism into the system in the future. Figure 1 gives the overall system architecture of the *Scene Editor*.

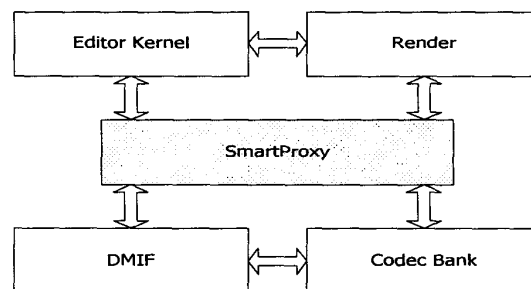


Figure 1. System architecture of *Scene Editor*

Modules Interoperation

To compose different media objects, some extra information called *scene description* [1,2] is required. Scene description includes temporal and spatial relationship among media objects, composition information, and the interactive information of scene. MPEG-4 specification proposes BIFS (Binary Format for Scene) to represent the scene description. The scene description mechanism allows users to describe and modify a multimedia content without getting involved with the encoding and decoding of the native data. To

playback the scene[5,7], *Editor Kernel* will invoke BIFS decoder to decode the scene stream from *DMIF* and then restore the original scene description tree. With this scene description tree, *Render* can compose other media streams from *Codec Bank* to generate the correct scene.

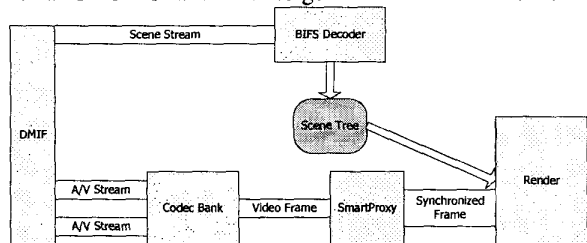


Figure 2. Flow of scene stream from *DMIF* to *Render*.

With the input and output interface of our *Codec Bank*, we can ignore the difference between source media data as described in Figure 3. The *SmartProxy* module is mainly for coordinating different decoders.

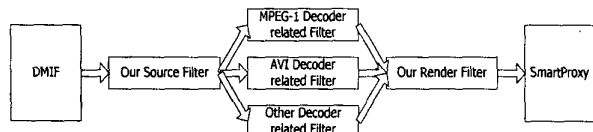


Figure 3. *Codec Bank* and *DMIF*, *SmartProxy*

System Implementation

We have implemented the scene editor on Microsoft Windows platform and demonstrated that the object-oriented concept of MPEG-4 is indeed feasible. The implementation environment is as follows:

Operating System	Microsoft Windows 98/Windows 2000
SDK Support	Microsoft DirectX 7.0 Microsoft DirectX Media 6.0
Compiler	Microsoft Visual Studio 6.0
CPU	Intel Pentium III-500

Figure 4 gives a typical snapshot of the scene editor. The left sub-window is the tree view of the scene description; the middle one is the preview window; and the right one is the property window.

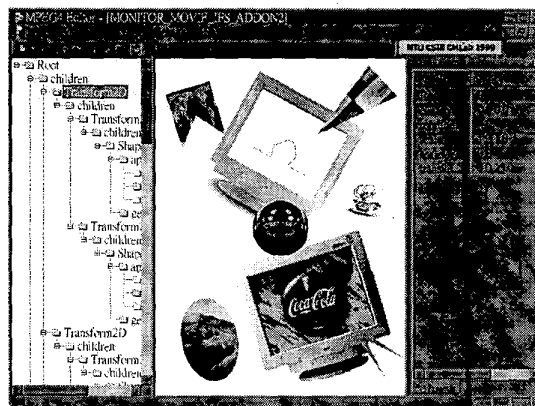


Figure 4. User interface of our MPEG-4 Scene Editor

Future Works

We expect to finish the following works in the second year of our MPEG-4 system development:

- In *DMIF*, our system needs to support remote media data access.
- The synchronization control is currently implemented in *Render* module. This however does not conform to the MPEG-4 specification. Besides, in our system the *Codec Bank* accesses and decodes data directly from *DMIF*. This approach is quite inefficient for the synchronization. We intend to move the load to *Editor Kernel* and thus follow the synchronization method described in the MPEG-4 specification.
- More decoder filters will be implemented and included in our *Codec Bank*.

Reference

- [1] ISO/IEC FDIS 14496, Information Technology – Generic Coding of Audio-Visual Objects – Part 1, Part 2, Part 3, Part 6, International Organization for Standardization, 1998.
- [2] International Standard ISO/IEC 14772-1:1997.
- [3] ISO/IEC IS 10918, Information Technology – Coding of Moving Pictures and Associated Audio for Digital Storage Media up to about 1.5 Mbit/s – Part 2, Part 3, International Organization for Standardization, 1991.
- [4] ISO/IEC IS 13818, Information Technology – Generic Coding of Moving Pictures and Associated Audio Information – Part 2, Part 3, International Organization for Standardization, 1994.
- [5] Video For Window, Microsoft Corporation.
- [6] DirectX Media, Microsoft Corporation, October 1998.
- [7] PC Video Synchronization and Playback, Microsoft Corporation, November 1998.