

장면 기술을 위한 MPEG-4 기반 XMT API 구현

정예선, 김규현, 기명석
한국전자통신연구원 무선방송연구소 방송미디어부
대전광역시 유성구 가정동 161번지

MPEG-4 based XMT APIs for Scene Description

Yesun Joung, Kyuheon Kim, Myungseok Ki
Broadcasting Media Technology Dept., Radio & Broadcasting Laboratory
Electronics and Telecommunications Research Institute
E-mail: ysjoung, kkim, serdong@etri.re.kr

요약

MPEG-4 시스템은 장면 자체를 하나의 구성 요소로 여기는 기존의 시스템과는 달리, 그 장면을 구성하는 부호화 또는 복호화된 A/V 객체(Audio/Visual Objects)들을 하나의 단위로 인식하여, 다양한 멀티미디어 콘텐츠의 장면을 구성(Scene Composition)하고 표현하는 것에 그 특징이 있다. 이러한 MPEG-4 시스템의 객체 기반 특징은 다양한 사용자와의 대화성(Interactivity)을 가능하게 하며, 또한 편리한 콘텐츠 편집 및 재사용등이 가능하기에 차세대 디지털 방송 콘텐츠 제작에 중요하게 활용될 전망이다. 객체 기반 A/V 편집 도구는 MPEG-4를 기반으로 차세대 디지털 방송 콘텐츠 제작을 용이하게 하기 위한 제작/편집 도구로써, 장면을 표현하기 위하여 BIFS(Binary Format for Scene description)와 XMT(eXtensible MPEG-4 Textual format) 포맷을 모두 사용하고 있다. BIFS 포맷은 저장된 결과물을 바이너리 형태로 표현하기 때문에, 저장된 결과물을 전송하는데에는 용이하나, 중간에 저장된 결과물을 확인하기 어렵고, 또한 기존의 다른 어플리케이션과의 상호 작용(Interoperability)과 교환(Exchange)에도 어려움이 따른다. 이에 반해, XMT는 차세대 마크업 언어로 각광 받고 있는 XML에 그 기반을 두고 있기에 저장된 결과물을 제작자가 쉽게 저작물을 이해할 수 있으며, SMIL과 X3D 같은 다른 어플리케이션과의 상호 작용과 교환 또한 용이하게 한다. XMT는 기술 방법에 따라 XMT-A와 XMT-O 두가지 형태가 있으며, XMT-A 포맷은 VRML에서 발전한 X3D(eXtensible 3D)를 바탕으로 MPEG-4 시스템의 특징들을 수용하여 구성되고 BIFS와 일대일로 대응된다. 반면에 XMT-O는 멀티미디어 문서를 웹문서로 표현하는 SMIL 2.0을 그 기반으로 하였기에 MPEG-4 시스템의 특징보다는 콘텐츠를 저장하는 제작자의 초점에 맞추어 개발된 형태이다. XMT를 이용하여 콘텐츠를 저장하기 위해서는 사용자 인터페이스를 통해 입력되는 저작 정보들을 손쉽게 저장하고 조작할 수 있으며, 또한 XMT 파일 형태로 출력하기 위한 API가 필요하다. 이에, 본 논문에서는 XMT 형태의 중간 자료형으로의 저장 및 조작을 위하여 XML에서 표준 인터페이스로 사용하고 있는 DOM(Document Object Model)을 기반으로 하여 XMT 문법에 적합하게 API를 정의하였으며, 또한, XMT 파일을 생성하기 위한 API를 구현하였다. 본 논문에서 제공된 API는 객체기반 제작/편집 도구에 응용되어 다양한 멀티미디어 콘텐츠 제작에 사용되었다.

1. 서론

오늘날과 같은 정보화 사회에서는 여러 가지 형태의 디지털화 된 서비스들, 즉 디지털 멀티미디어 서비스가 정보 통신망을 통하여 서로 유기적으로 결합되어 우리의 일상 생활에 이용될 것이다. 따라서, 지

금까지 서로 다른 영역으로 나뉘어져 있던 통신, 방송 및 컴퓨터 산업들 사이의 경계가 허물어지고 서로가 통합되고 융합되어 복합적이고 다양한 서비스들이 등장할 것이다. 동영상 및 통신 기능이 컴퓨터에 추가되고, 통신망이나 컴퓨터 망에 의하여 방송은 양방향성을 지향하고 있으며, 또한 무선 멀티미디어 서비스에 대한 수요도 날로 증가하고 있다. 정보화 사회에서는 원하는 정보를 언제, 어디서나, 누구와도 주고 받을 수 있으며, 이에 더하여 사용자가 원하고 다루기 편한 형태로 모든 정보를 가공할 수 있어야 한다. 따라서, 정보화 사회를 위한 멀티미디어 서비스는 기본적으로 통신, 방송 및 컴퓨터가 융합되어 멀티미디어 데이터의 편리한 조작 및 가공, 양방향성에 기반한 사용자와의 대화성(interactivity)이 강조되는 형태를 가질 수 밖에 없으며, 이러한 요구를 만족시키기 위해서는 데이터 압축 기술이외에 새로운 방향의 멀티미디어 부호화 기술이 요구되고 있다. 멀티미디어 데이터의 편리한 조작 및 가공, 사용자와의 대화성을 제공하기 위해서는 현재의 단순한 압축 기술로는 한계가 있기에, 멀티미디어 데이터를 조작하고 가공하기 쉬운 형태로 분할하여 부호화하는 객체기반의 부호화 기술이 필요하게 되었다. 따라서, 오디오 및 동영상 부호화 기술의 국제 표준을 제정하고 있는 MPEG(Moving Pictures Experts Group)에서는 위와 같은 요구를 수용하여 객체기반의 멀티미디어 부호화 기술인 MPEG-4를 표준화 하였다. 기본적으로 MPEG-4 표준안은 시스템(systems), 비주얼(visual), 오디오(audio), 적합성 시험(conformance testing), 참조 소프트웨어(reference software) 멀티미디어 전달을 위한 통합구조(DMIF: Delivery Multimedia Integration Framework)의 6개 부분(parts)을 표준화 하였으며 현재는 다양한 요구를 수용하기 위한 표준화 작업을 하고 있다.

이중에서 MPEG-4 시스템은 객체 단위로 부호화되고 복호화된 A/V 객체(Audio/Visual Objects)들을 합성하여 하나의 장면을 구성(scene composition)하고, 이러한 장면의 조합에 의해 구성된 멀티미디어 콘텐츠를 표현(multimedia content representation)하는데 그 목적이 있다. 이러한 장면을 장면 그래프(scene graph)라는 것으로 표현하고, 또한 BIFS(Binary Format for Scene description)를 이용하여 기술한다. 그러나 BIFS 포맷의 경우, 바이너리 포맷이기 때문에 저장된 결과물을 중

간에 확인하기가 어렵고, 다른 어플리케이션과의 상호 작용과 교환에 어려운 점이 있다. 이런 점을 보완하여, MPEG-4 장면 기술을 사용자에게 친근하고 다른 어플리케이션과의 상호작용 (interoperability)과 교환 (exchanging)을 가능하게 하는 텍스처 포맷, XMT (eXtensible MPEG-4 Textual format)이 제공되었다. XMT는 XML에 기반한 텍스처 포맷으로 XMT-A와 XMT-O의 두가지가 있다. XMT-A 포맷은 VRML (Virtual Reality Markup Language)에서 발전한 X3D (eXtensible 3D)를 바탕으로 MPEG-4의 특징을 수용하여 구성되어 있으며, BIFS와 일대일로 대응된다. 또한, XMT-O는 XML형태로 상호 작용이 가능한 멀티미디어 문서를 표현하는 SMIL 2.0을 바탕으로 구성되어있기에, XMT-A와는 달리 복잡한 MPEG-4 특징들을 일대일로 표현하기 보다는 콘텐츠를 저작하는 저작자에 초점이 맞추어진 포맷이다.

본 논문에서는 현재 개발중인 MPEG-4 콘텐츠 저작도구에 대한 소개, 콘텐츠를 저작할 때 XMT로 나타내기 위해 필요한 API에 대한 소개, 구현된 예를 소개하고 결론을 맺고자 한다.

2. 객체기반 A/V 편집/저작 도구

구현된 저작 도구의 전체적 구조는 [그림 1]과 같이 크게 사용자 인터페이스부, 편집 정보 처리부, 미디어 라이브러리의 세부분으로 나눌 수 있다.

사용자 인터페이스 부는 그래픽 인터페이스 모듈과 Data Access API 모듈로 구성되며, 콘텐츠를 저작하고자 하는 제작자에게 편리하게 MPEG-4 표준에 상응하는 콘텐츠를 저작 할 수 있는 인터페이스를 제공한다. Data Access API 모듈은 그래픽 인터페이스와 다른 모듈들을 연결하는 기능을 가지고 있다. 예를 들어, 각각의 하위 모듈들은 Data Access API를 통하여 그래픽 인터페이스에게 해당 모듈의 편집 정보나 데이터를 주고 받는다.

편집 정보 처리부는 XMT-A/O API 모듈, XMT Mapper 모듈과 MP4 컨버터 모듈로 구성된다. 사용자 인터페이스부를 통하여 입력되는 콘텐츠 편집 정보들을 DOM 내부 데이터 구조로 저장하고 관리하는 기능들은 XMT-A/O API 모듈에서 담당한다. XMT Mapper 모듈은 XMT-O를 BIFS와 일대일 매칭이 되는 XMT-A로 바꿔주는 기능을 하며, 마지막으로 MP4 컨버터 모듈은 XMT-O/A API 모듈을 통해 생성된 XMT를 MP4 파일로 변환 (translation) 한다. 이렇게 생성된 MP4 파일은 MPEG-4 player에서 동작하게 된다.

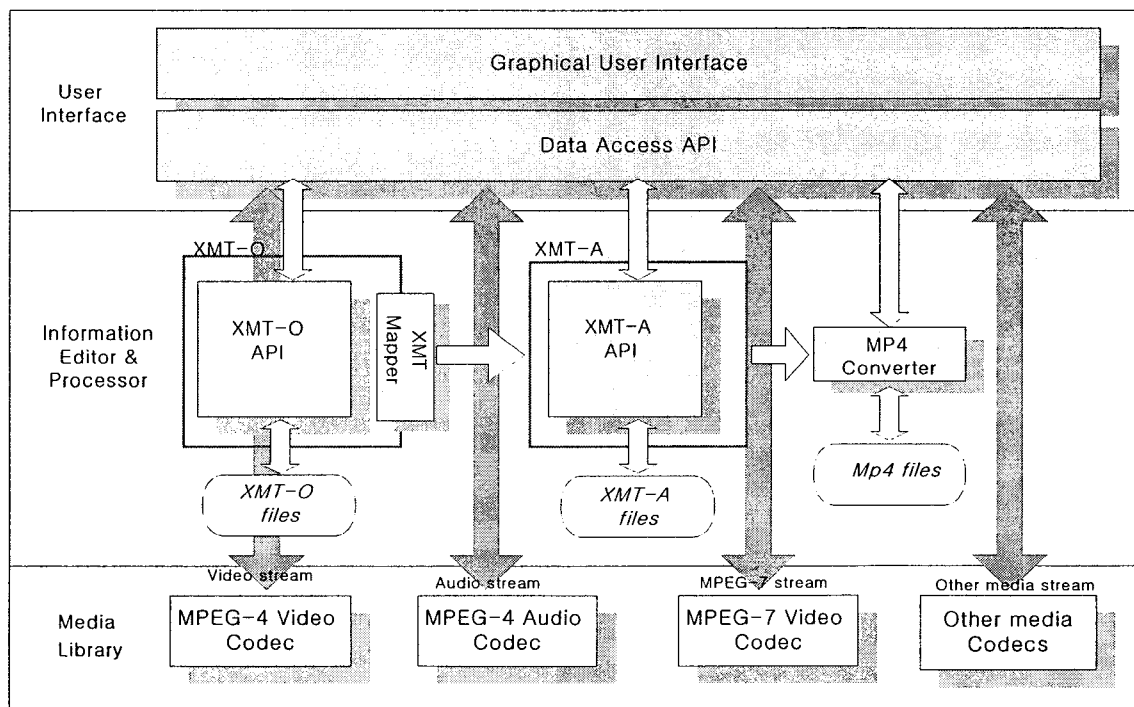
미디어 라이브러리는 오디오/비주얼 객체들을 부호화 또는 복호화 하기 위해 다양한 미디어 코덱들을 제공하며, 각각의 코덱 모듈들은 Data Access API를 통하여 사용자 인터페이스와 동작한다.

3. eXtensible MPEG-4 Textual format

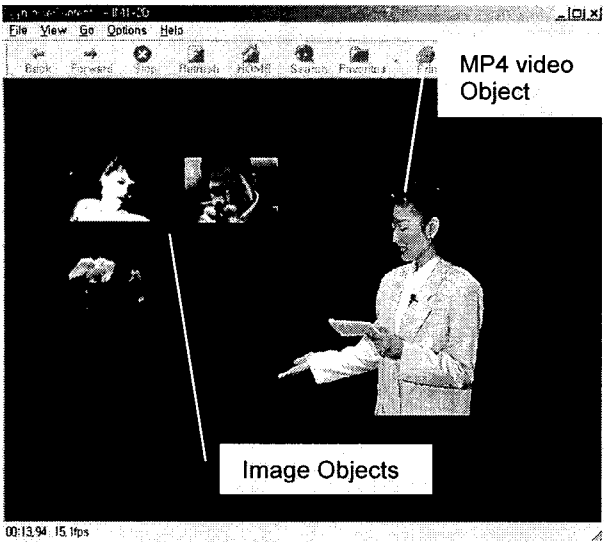
eXtensible MPEG-4 Textual format (XMT)은 MPEG-4 장면 그래프를 textual 형식으로 표현하기 위한 framework 로써, XMT는 콘텐츠 저작자들이 자신의 콘텐츠를 다른 저작자 또는 서비스 제공자와 상호 교환 (exchanging)할 수 있도록 해주며, X3D, SMIL과의 상호 운영 (interoperability)이 용이하도록 정의 되었다. XMT 포맷은 XMT-A 프로파일과 XMT-O 프로파일 두가지가 있으며, 본 논문에서는 [그림 2]와 같이 본 저작툴에서 저작된 콘텐츠가 상기에서 설명된 두가지 프로파일로 어떻게 구성되는지 예시함으로써 두 프로파일에 대하여 설명하고자 한다. 예제는 영화 상영작을 안내하는 콘텐츠로, 임의 형상의 MP4 video 객체하나와 3개의 Image 객체로 구성되어 있다.

3.1 XMT-A 프로파일

XMT-A 프로파일은 MPEG-4의 장면정보를 XML 형태의 텍스처로 표현하기 때문에 MPEG-4의 장면정보를 바이너리 형태로 표현하는 BIFS와는 일대일로 대응된다. [그림 3~5]에서 예제 콘텐츠를 XMT-A로 나



[그림 1] MPEG-4 저작 도구 구조



[그림 2] 예제 콘텐츠

타 내었다. [그림 3]은 XMT-A 파일의 구조를 보여주고 있는데, “+”는 하위에 description 이 존재하는 것이고, “-”는 하위에 description 이 존재하지 않음을 나타낸다. 즉, 예를 들어 마지막 Transform2D 노드에 대한 하위 description 은 [그림 4]에서와 같이 해당 노드에 대한 설명이 나타나 있다. 구조적으로 XMT-A 는 header 와 body 로 구성되어 있으며, header 에는 InitialObjectDescriptor 정보와 저작과 관련된 meta 정보, 즉 저작자, 저작 날짜와 같은 사항이 존재한다. body 에는 scene 정보와 scene 을 구성하는 미디어 객체에 대한 정보들이 ObjectDescriptor 로 나타난다. [그림 4]는 장면을 구성하는 비디오 객체의 화면상 정보를 XMT-A 로 설명한 그림이며, [그림 5]는 장면을 구성하고 있는 비디오 객체에 대한 특징을 설명하는 것으로서, 일례로 비디오 파일의 위치, encoding hints, decoding information 등의 정보가 존재한다.

```

- <Header>
+ <InitialObjectDescriptor ObjectDescriptorID="0">
  </Header>
- <Body>
- <Replace>
- <Scene>
- <Group>
- <children>
+ <Transform2D DEF="Trans0" translation="-201 121">
+ <Transform2D DEF="Trans1" translation="-202 25">
+ <Transform2D DEF="Trans2" translation="-77 120">
+ <Transform2D DEF="Trans3" translation="33 21">
</children>
</Group>
</Scene>
</Replace>
- <Par begin="0.0">
- <ObjectDescriptorUpdate>
+ <OD>
+ <OD>
+ <OD>
+ <OD>
</ObjectDescriptorUpdate>
</Par>
</Body>
</XMT-A>

```

[그림 3] 예제 콘텐츠에 대한 XMT-A 파일

```

+ <Transform2D DEF="Trans3" translation="33 21">
- <children>
- <Shape DEF="Video3">
- <geometry> <Bitmap scale="352 288" /> </geometry>
- <appearance>
- <Appearance>
- <material>
- <Material2D>
- <lineProps> <LineProperties /> </lineProps>
</Material2D>
</material>
- <texture>
  <MovieTexture DEF="Texture3" url="od6" repeatS="FALSE"
    repeatT="FALSE" stopTime="-1" startTime="0" />
</texture>
</Appearance>
</appearance>
</Shape>
</children>
</Transform2D>

```

[그림 4] 예제 콘텐츠에 대한 XMT-A 파일 (화면구성상)

```

+ <OD>
- <ObjectDescriptor ObjectDescriptorID="od6">
- <Descr>
- <esDescr>
- <ES_Descriptor ES_ID="es6">
- <StreamSource url="file:///IF_weather_cif.m4v">
  <EncodingHints streamFormat="MPEG4-VideoFDIS" />
</StreamSource>
- <decConfigDescr>
- <UserDecoderConfigDescriptor streamType="4" bufferSizeDB=
  "4600000" objectTypeIndication="0x22">
- <decSpecificInfo>
  <UserDecoderSpecificInfo customData="352 288 Binary" />
</decSpecificInfo>
</UserDecoderConfigDescriptor>
</decConfigDescr>
- <slConfigDescr>
<SLConfigDescriptor OCRResolution="1000"
  AU_seqNumLength="8" timeStampLength="16"
  useTimeStampsFlag="True" packetSeqNumLength="8"
  timeStampResolution="1000" useAccessUnitEndFlag="True"
  useAccessUnitStartFlag="True"
  useRandomAccessPointFlag="True" />
</slConfigDescr>
</ES_Descriptor>
</esDescr>
</Descr>
</ObjectDescriptor>
</OD>

```

[그림 5] 예제 콘텐츠에 대한 XMT-A 파일 (객체특징)

3.2 XMT-O 프로파일

XMT-O 는 SMIL 에 기반을 두고 있으며, MPEG-4 특성에 대한 high-level 적 접근이다. 그러므로, XMT-O 는 Audio/Visual 객체와 그들의 관계를 high-level 에서 묘사하기 때문에 콘텐츠에 대한 요구사항을 노드와 라우트 연결에 대한 명확한 코딩으로 나타내기 보다는 저작자의 의도에 의해 표현할 수 있다. 예를 들어, [그림 3~5]에서 보듯 XMT-A 의 포맷인 경우 일반적인 콘텐츠 저작자가 사용하기는 어려운 InitialObjectDescriptor, encoding hints, decoding information 등의 MPEG-4 에 관련된 description 이 존재하지만, [그림 6]에서 나타나듯이 XMT-O 는 img 또는 video 와 같은 노드명을 사용함으로써 저작자가 보다 쉽게 콘텐츠를 저작할 수 있도록 구성되었다. 그림에서 보듯이 XMT-O 역시 head 와 body 로 구성이 되며, head 의 경우 콘텐츠

의 크기를 지정하고, meta 정보를 표현할수 있으며, body 부분에는 장면 정보를 나타낼수 있다.

```
<?xml version="1.0" encoding="UTF-8" ?>
-<XMT-O>
-<head>
-<layout type="xmt/xmt-basic-layout" metrics="pixel">
  <topLayout height="400" width="600" />
</layout>
</head>
-<body>
-<par begin="0ms">
+ 
+ 
+ 
+ <video id="Video4" src="file:///IF_weather_cif.m4v">
</par>
</body>
</XMT-O>
```

[그림 6] 예제 콘텐츠에 대한 XMT-O 파일

4. XMT APIs 구현

구현된 XMT APIs 의 구조는 [그림 7]에서와 같이 XMT Manager APIs, XMT base APIs 들, DOM 내부자료 구조의 소프트웨어 모듈로 나눌 수 있다.

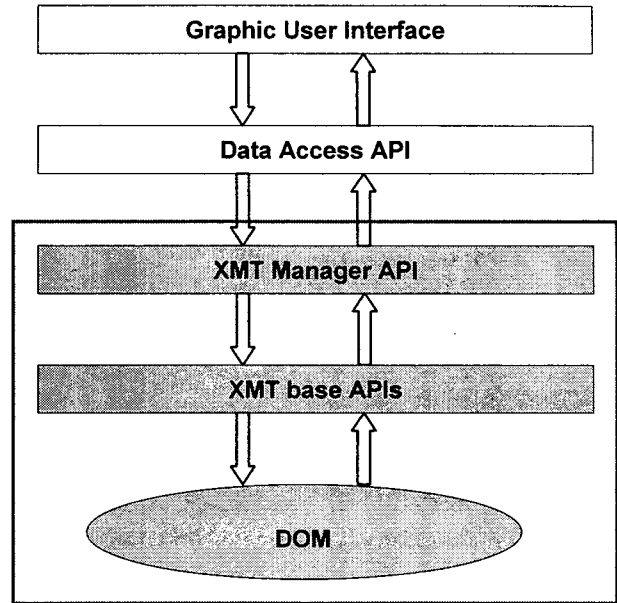
첫번째, XMT Manager APIs 부분은 사용자 인터페이스 모듈의 Data Access APIs 부분과 결합되는 부분으로 사용자 인터페이스부의 그래픽 인터페이스(GUI)를 통하여 입력되는 화면 구성 객체에 대한 정보를 XMT 의 구성 노드별로 분리하고, 또한 역으로, XMT base APIs 로부터 받은 노드별 정보를 하나의 화면 구성 객체로 Data Access APIs 에 전달하는 API 들로 구성된다.

두번째, XMT base APIs 부분은 XMT-A 와 XMT-O 프로파일에 맞게 DOM(Document Object Model) 형태의 내부 자료구조에 해당 노드별 속성 정보를 저장하기 위한 모듈이다. IBM 의 XML for C++의 DOM Interface 를 사용하여 XMT 를 저장할 수 있는 APIs 를 구성하였다.

세번째, 내부적인 DOM 자료 구조는 입력된 사용자 편집정보와 관련된 속성 정보나 로딩된 장면관련 속성 정보들을 저장하는 소프트웨어 모듈이다. DOM 은 XML 문서를 사용하기 위한 표준화된 인터페이스로 본 논문에서는 DOM level2를 기반으로 구현하였다.

5. 결론

객체기반 차세대 디지털 방송 콘텐츠 제작을 위하여 구현된 객체기반 A/V 편집도구는 보다 다양한 객체기반 특징을 표현하기 위하여 MPEG-4 표준을 사용하였다. 이에, 장면을 표현하기 위하여 MPEG-4 에서 정의한 BIFS(Binary Format for Scene description)와 XMT(eXtensible MPEG-4 Textual format) 포맷을 모두 사용하고 있다. 바이너리 형태인 BIFS 포맷은 저장된 결과물을 전송하는데에 용이하며, 이에 반하여, XMT 는 텍스트형 포맷으로 구성되어 있기에 사용자가 쉽게 이해 할 수 있으며, 타 어플리케이션과도 쉽게 교환 할 수 있는 장점들이 있다. 또한, XMT 를 이용하여



[그림 7] XMT API 소프트웨어 구조도

컨텐츠를 저작하기 위해서는 사용자 인터페이스를 통해 입력되는 저작 정보들을 저장/조작할 수 있고, 또한 XMT 파일 형태로 출력하기 위한 API 가 필요하다. 이에, 본 논문에서는 XMT 형태의 중간 자료형으로의 저장 및 조작을 위하여 XML 에서 표준 인터페이스로 사용하고 있는 DOM 을 기반으로 하여 XMT 문법에 적합하게 XMT 정보를 저장/조작 할 수 있는 API 를 정의하였으며, 또한, XMT 파일을 생성 및 로딩하기 위한 API 를 구현하였다.

본 논문에서 제공된 API 는 객체기반 제작/편집 도구에 응용되어 다양한 멀티미디어 콘텐츠 제작에 사용되었다.

Acknowledgement

본 연구는 정보통신부의 “객체기반 AV 편집/제작 도구 개발”사업의 일환으로 수행된 연구 결과입니다.

참고 문헌

- [1] ISO/IEC JTC 1/SC 29/WG11 M7496, Study of ISO/IEC 14496-1:2001 PDAM2 July 2001.
- [2] ISO/IEC 14496-1:2001 PDAM2 (MPEG-4 Systems), Singapore, March 2001.
- [3] ISO/IEC 14496-1:1999 FDAM1 (MPEG-4 Systems), Maui, December 1999.
- [4] W3C recommendation(Ver. 1.0), Document Object Model (DOM) Level 2 Core Specification, <http://www.w3c.org/TR/DOM-Level-2-Core>.
- [5] Web3D Consortium, VRML 200x work in progress, <http://www.w3c.org/TaskGroups/x3d/specification>.
- [6] Synchronized Multimedia Integration Language (SMIL) 2.0 specification, <http://www.w3c.org/TR/2000/WD-smil20-20000921>.