

XMT-A를 이용한 MPEG-4 콘텐츠 저작

Authoring of MPEG-4 Contents using XMT-A

이법기

기술신용보증기금 차장

정원식

한국전자통신연구원 선임연구원

고일석

충북과학대학 전자상거래 교수

최영수

한국원자력연구소 선임연구원

한찬호

경운대학교 소프트웨어공학과 교수

Bub-Ki Lee

Deputy chief, Korea Technology Credit Guarantee Fund

Won-Sik Cheong

Senior Engineer, Electronics and Telecommunications Research institute

Il-Seik Ko

Professor, Chungbuk Provincial College of Science & Technology

Young-Soo Choi

Senior Engineer, Korea Atomic Energy Research Institute

Chan-Ho Han

Professor, Kyungwon University

중심어 : MPEG-4, XMT, MP4

요약

다양한 형태의 차세대 오디오/비디오 서비스를 위한 멀티미디어 데이터를 유연성 있게 부호화하기 위한 국제 표준 규격인 MPEG-4에서는 저작된 멀티미디어 콘텐츠를 저장하기 위한 포맷으로 XMT (eXtensible MPEG-4 Textual format)와 MPEG-4 파일 포맷 (MP4)를 제공하고 있다. 이들 중 XMT 포맷은 사람이 콘텐츠의 내용을 읽을 수 있는 가독성을 가지고 있는 장점이 있으며, XML을 기반으로 하고 있기 때문에 여러 가지 응용에 사용될 수 있다. 본 논문에서는 인터랙티브 멀티미디어 콘텐츠를 저작할 수 있는 국제 표준인 MPEG-4를 따르는 콘텐츠를 XMT-A를 이용하여 저작 하는 방법에 대하여 기술한다.

Abstract

MPEG-4 is an international standard for interactive multimedia data that is used for next generation audio-visual service. It presents an two types of storage format such as XMT (eXtensible MPEG-4 Textual format) and MP4 file format to store the interactive multimedia contents. The XMT, a textual format, has advantage of readability and it can be used in many applications since designed based on the XML. In this paper, we present the way how to author the MPEG-4 contents using XMT.

1. 서론

Extensible MPEG-4 Textual Format (XMT)는 MPEG-4 장면 기술을 textual 형식으로 표현하기 위한 framework이다. XMT는 콘텐츠 저작자들이 자신의 콘텐츠를 다른 저작자, 도구 또는 서비스 제공자와 상호 교환할 수 있도록 해주며, X3D, SMIL과의 상호 운영이 용이하도록 해준다.

XMT framework은 XMT-A 포맷과 XMT-U 포맷의 두가지 레벨이 있다. XMT-A는 MPEG-4 콘텐츠의 XML 기반 버전

으로서 X3D의 subset과 MPEG-4의 특성을 표현하기위한 X3D의 확장으로 구성되어 있으며, 이진 포맷과 textual 포맷간의 일대일 대응을 제공한다. 반면에 XMT-U는 SMIL에 기반을 두고 있으며, MPEG-4 특성에 대한 high-level 추상화이다. 그러므로 XMT-U는 오디오-비주얼 객체와 그들의 관계를 high-level에서 묘사하기 때문에 콘텐츠에 대한 요구 사항을 노드와 라우트 연결에 대한 명확한 코딩으로 나타내기 보다는 저작자의 의도에 의해 표현할 수 있다.

본 논문에서는 XMT-A를 이용한 MPEG-4 콘텐츠의 저작 방법에 대하여 기술한다. 먼저, 본 논문에서는 MPEG-4에서

지원하는 다양한 미디어 객체, 즉 오디오, 비디오, 이미지, 텍스트 및 2차원 그래픽 객체들이 XMT-A에서 어떻게 표현 되는지 살펴보고자 한다.

II. XMT-A 문서의 구조

XMT-A 문서는 그림 1에 나타나 있는 것과 같이, 하나의 <Header> 엘리먼트 (element)와 하나의 <Body> 엘리먼트로 구성된다. 먼저, <Header> 엘리먼트는 선택적인 엘리먼트로서 영 또는 하나 이상의 <meta> 엘리먼트를 가질 수 있으며, MPEG-4 엘리먼트중의 하나인 <InitialObjectDescriptor> 엘리먼트를 가진다. 여기서, <meta> 엘리먼트는 XMT 문서에 대한 메타 데이터를 기술하기 위한 엘리먼트이고, <InitialObjectDescriptor>는 MPEG-4 콘텐츠를 play하기 위하여 필요한 초기화 정보를 담고 있는 엘리먼트이다. 그리고, <Body> 엘리먼트는 <Replace>와 <Scene> 엘리먼트를 가지며, 여기에 장면의 내용을 기술 한다.

```

<Header>
  <meta>
</meta>
  <InitialObjectDescriptor/>
</Header>
<Body>
  <Replace>
    <Scene>
      <!-- The scene contents -->
    </Scene>
  </Replace>
</Body>
    
```

그림 1. XMT-A 문서의 기본 구조

III. XMT-A를 이용한 MPEG-4 콘텐츠의 저작

본 절에서는 XMT-A를 이용한 MPEG-4 콘텐츠의 저작 방법을 그림 2에서 보는 바와 같이 비디오, 오디오, 이미지, 텍스트를 포함하는 간단한 MPEG-4 콘텐츠의 저작 예를 이

용하여 살펴보고자 한다. 이 콘텐츠는 하나의 비디오, 하나의 오디오, 하나의 텍스트 객체 및 그림에는 나타나있지 않지만 하나의 오디오 객체를 포함하는 것이다.

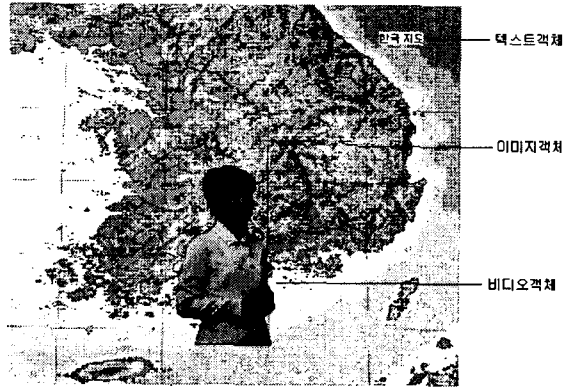


그림 2. 간단한 MPEG-4 콘텐츠의 예.

1. 예에 대한 XMT-A 파일의 전체 구성

XMT-A 파일로 만들어진 그림 2에서의 콘텐츠를 표현하는 XMT-A 파일의 전체 구성을 그림 3에 나타내었다. 이 그림은 XMT-A 파일을 마이크로소프트의 인터넷 익스플로러 (internet explorer)에서 본 것으로, XMT 파일은 XML로 기술되기 때문에 인터넷 익스플로러에서 볼 수 있다. 또한, 그림에서 +로 표시된 노드 엘리먼트는 그 아래에 자식 노드 엘리먼트들이 있다는 것을 나타낸다. 이 그림에서는 XMT-A 파일의 전체 구성을 보기 위하여 상위 노드 엘리먼트들만을 나타내었다.

이 그림을 살펴보면, 그림 1에서 나타낸 바와 같이 <Header>, <Body>로 구성된 기본 골격을 가지고 있으며, <Header>에는 <InitialObjectDescriptor> 엘리먼트만 가지고 있다. 그리고, <Body> 부분은 크게 <Replace><Scene> (line 006,007) 부분과 <Par> (line 018) 부분으로 나눌 수 있다. 이 예에서 <Replace><Scene> 부분에는 장면 기술이 포함되어 있고, <Par> 부분에는 객체 기술 (Object description)이 포함되어 있다.

line 008을 살펴보면 <Scene> 노드 엘리먼트의 자식 노드로는 MPEG-4 시스템스 규격에 SFTop 노드로 정의되어 있는 네 가지의 노드 중 하나가 올 수 있으며, 이 예제에서는 <Group> 노드 엘리먼트를 사용하였다. 그리고, <Group> 노드 엘리먼트의 child 노드로 4개의 <Transfor2D> 노드가

있으며, 이들은 각각 예제에서 사용된 4개의 미디어 객체에 대응된다. 이 노드 엘리먼트의 속성으로 사용된 DEF는 이 노드 엘리먼트가 다른 곳에서 참조 또는 재 사용될 경우에 이 노드 엘리먼트의 이름으로 사용된다. 또한 translation 필드의 경우에는 미디어 객체의 공간적인 위치를 나타낸다. 이때 화면의 중앙의 좌표가 "0 0"이 되며, <Transform2D> 노드가 다른 노드의 자식 노드로 사용된 경우에는 부모 노드의 좌표에 대한 상대적인 좌표를 나타낸다.

```

001 - <XMT-A>
002 - <Header>
003 + <InitialObjectDescriptor ObjectDescriptorID="0">
004 </Header>
005 - <Body>
006 - <Replace>
007 - <Scene>
008 - <Group>
009 - <children>
010 + <Transform2D DEF="Trans5" translation="29 -35">
011 + <Transform2D DEF="Trans10002" translation="169 158">
012 + <Transform2D DEF="Trans6" translation="-57 -90">
013 + <Transform2D DEF="Trans7" translation="-296 216">
014 </children>
015 </Group>
016 </Scene>
017 </Replace>
018 - <Par begin="0.0">
019 - <ObjectDescriptorUpdate>
020 + <OD>
021 + <OD>
022 + <OD>
023 </ObjectDescriptorUpdate>
024 </Par>
025 </Body>
026 </XMT-A>
    
```

그림 3. 그림 1에 나타난 예에 대한 XMT-A 파일의 전체 구성

line 18에 나타나 있는 <Par> 노드 엘리먼트를 살펴보면, 자식 노드로 <ObjectDescriptorUpdate> 노드, 이의 자식 노드로는 <OD> 노드가 사용되고 있다. <Par> 노드는 XMT-A 포맷에서 동기화를 담당하고 있는 노드로서, 여기서는 위의 <Replace> 노드와 동시에 실행된다는 뜻으로 사용되어, 예제에서 사용된 비디오, 오디오, 이미지 객체에 대한 객체 기술, 즉 <OD> 정보를 객체 기술 명령 (Object Descriptor command)인 <ObjectDescriptorUpdate>를 통하여 나타내는 역할을 하고 있다. 여기서, 텍스트 객체의 경우에는 장면 기술에서 텍스트 객체의 모든 속성을 나타낼 수 있기 때문에 텍스트 객체에 대한 <OD> 노드 엘리먼트는 존재하지 않는다.

2. 이미지 객체의 구성

XMT-A 를 이용하여 이미지 객체를 표현하기 위해서는 장면 기술과 객체 기술이 필요하다. 또한, 이미지 객체에 대한 표현은 많은 노드 엘리먼트의 조합으로 이루어지며, 아래에 제시될 예제는 이미지 객체를 표현하기 위한 하나의 방법이며, 이외에도 다른 노드 엘리먼트의 조합으로도 나타낼 수 있다.

2.1. 장면 기술 부분

이미지 객체에 대한 XMT-A에서의 장면 기술 부분 (그림 3의 line 010)을 그림 4에 나타내었다. 이미지 객체에 대한 최상위 노드로는 <Transform2D> 노드가 사용되며, 자식 노드로는 장면의 모양을 나타내는 <Shape> 노드가 사용되며, 이의 자식 노드로 <geometry>와 <appearance>가 사용되었다.

여기서, <geometry> 노드는 객체의 기하학적인 모양을 나타내고, 자식 노드인 <Bitmap> 노드는 객체가 비트맵임을 나타내며, 속성인 scale은 비트맵의 크기를 나타낸다. 또한, <appearance> 노드는 객체의 외형, 예를 들면 선 색, 채우기 색, 선 굵기, 텍스트 종류 등을 나타낼 수 있는 노드로서, 이 예에서는 이미지 객체이기 때문에 다른 외형에 대한 정보는 필요 없고, 텍스트로 이미지를 사용한다는 정보로 <ImageTexture>를 사용하였다. 여기서 속성으로 사용된 url은 이미지 객체에 대한 객체 기술자 (object descriptor)의 ID를 나타낸다.

```

001 - <Transform2D DEF="Trans5" translation="29 -35">
002 - <children>
003 - <Shape DEF="Image5">
004 - <geometry>
005 <Bitmap scale="640 480" />
006 </geometry>
007 - <appearance>
008 - <Appearance>
009 - <texture>
010 <ImageTexture DEF="Texture5" url="005" />
011 </texture>
012 </Appearance>
013 </appearance>
014 </Shape>
015 </children>
016 </Transform2D>
    
```

그림 4. 이미지 객체에 대한 장면 기술 부분

2.2 객체 기술 부분

이미지 객체에 대한 객체 기술 부분 객체 기술자는 하나 이상의 기초 스트림 기술자 (elementary stream (ES) descriptor)를 포함하고 있으며, <ES_Descriptor>는 <StreamSource>, <DecoderConfigDescriptor>, <SLConfigDescriptor>를 포함한다. 이들에 대하여 기술하면 다음과 같다. 그림 5는 이미지 객체에 대한 객체 기술을 나타내었다.

- <ES_Descriptor>
 - ▶ 기초 스트림 기술자를 나타냄
 - ▶ ES_ID: 기초 스트림의 ID를 나타냄
- <StreamSource>
 - ▶ 기초 스트림이 저장되어 있는 위치를 나타내기 위한 노드 엘리먼트
 - ▶ url: 기초 스트림이 위치해 있는 url을 나타냄
 - ▶ <Encoding Hints>
 - 일반적으로는 압축되지 않은 비디오 또는 이미지 데이터를 비트스트림으로, 예를 들면 압축되지 않은 비디오 시퀀스를 MPEG-4 비디오로 부호화하기 위해서 필요한 파라미터들을 지정하기 위해서 사용하는 노드 엘리먼트임.
 - 이 예제에서는 현재 기초 스트림 기술자가 나타내고 있는 것이 JPEG으로 부호화된 정지영상임을 나타내기 위하여 사용되었음.
 - streamFormat
 - 현재 스트림의 포맷을 나타내기 위한 속성 임
 - 사용 할 수 있는 streamFormat 으로는 JPEG, MPEG4-Video, G723 (G.723 audio), AAC (MPEG-4 AAC audio), H263 (H.263 video), BIFS (OD 또는 BIFS stream) 등이 있음.
- <DecoderConfigDescriptor>
 - ▶ 기초 스트림을 복호화 하기위해서 필요한 정보들을 나타내기 위한 노드 엘리먼트임
 - ▶ streamType
 - 기초 스트림의 타입, 즉 비디오 스트림이나 오디오 스트림이나 등을 나타냄
 - 위의 예에서 streamType="4"는 비주얼 스트림임을 나타냄
 - 스트림 타입 값들을 표 1에 나타내었다.

```

001 - <CD>
002 - <ObjectDescriptor ObjectDescriptorID="cd8">
003 - <Desc>
004 - <ES_Desc>
005 - <ES_Descriptor ES_ID="es8">
006 - <StreamSource url="files//asp.jpg">
007 - <EncodingHints streamFormat="JPEG" />
008 - </StreamSource>
009 - <DecConfigDesc>
010 - <DecoderConfigDescriptor streamType="4"
011 - bufferSizeDB="46000"
012 - objectTypeIndication="0x6C">
013 - <decSpecificInfo
014 - <UseDecoderSpecificInfo customData="obsolete
015 - starting" />
016 - </decSpecificInfo>
017 - </DecoderConfigDescriptor>
018 - </DecConfigDesc>
019 - <SLConfigDesc>
020 - <SLConfigDescriptor COResolution="1000"
021 - AU_segNumLength="8"
022 - timeStampLength="16"
023 - useTimeStampsFlag="True"
024 - packetSeqNumLength="8"
025 - timeStampResolution="1000"
026 - useAccessUnitEndFlag="True"
027 - useAccessUnitStartFlag="True"
028 - useRandomAccessPointFlag="True" />
029 - </SLConfigDesc>
030 - </ES_Descriptor>
031 - </esDesc>
032 - </Desc>
033 - </ObjectDescriptor>
034 </CD>
    
```

그림 5. 이미지 객체에 대한 객체 기술 부분

표 1. streamType 값들

streamType value	stream type description
0x00	Forbidden
0x01	ObjectDescriptorStream
0x02	ClockReferenceStream
0x03	SceneDescriptionStream
0x04	VisualStream
0x05	AudioStream
0x06	MPEG7Stream
0x07	IPMPStream
0x08	ObjectContentInfoStream
0x09	MPPEGStream
0x0A - 0x1F	reserved for ISO use
0x20 - 0x3F	user private

- ▶ bufferSizeDB
 - 기초 스트림을 복호화 하기 위해서 필요한 버퍼의 크기를 바이트 수로 나타냄
- ▶ objectTypeIndication
 - 객체 또는 장면의 타입을 나타냄.
 - 위의 예에서 objectTypeIndication="0x6C"는 JPEG (Visual ISO/IEC 10918-1)임을 나타냄
 - 스트림 타입 값들을 표 2에 나타내었다.

표 2. objectTypeIndication 값들

Value	ObjectTypeIndicationDescription
0x00	Forbidden
0x01	SystemISOIEC14496-1
0x02	SystemISOIEC14496-1 AMD1
0x03-0x1F	reserved for ISO use
0x20	Visual ISOIEC14496-2
0x21-0x3F	reserved for ISO use
0x40	Audio ISOIEC14496-3
0x41-0x5F	reserved for ISO use
0x60	Visual ISOIEC13818-2 Simple Profile
0x61	Visual ISOIEC13818-2 Main Profile
0x62	Visual ISOIEC13818-2 SNR Profile
0x63	Visual ISOIEC13818-2 Spatial Profile
0x64	Visual ISOIEC13818-2 High Profile
0x65	Visual ISOIEC13818-2 422 Profile
0x66	Audio ISOIEC13818-7 Main Profile
0x67	Audio ISOIEC13818-7 Low Complexity Profile
0x68	Audio ISOIEC13818-7 SSR Profile
0x69	Audio ISOIEC13818-3
0x6A	Visual ISOIEC11172-2
0x6B	Audio ISOIEC11172-3
0x6C	Visual ISOIEC10918-1
0x6D-0xEF	reserved for ISO use
0x0D-0xFE	user private
0xFF	no object type specified

▶ <decConfigDescr>

- 각 객체 타입에 따라 특정 미디어 복화화에 필요한 정보들을 표현하기 위하여 사용되는 정보임
- 여기서는 JPEG의 경우 특별한 정보가 필요 없다는 뜻에서 line 14를 사용하였음.

● <SLConfigDescriptor>

- ▶ 기초 스트림의 sync layer header의 configuration을 정의하기 위하여 사용 됨.

3. 비디오 객체의 구성

3.1. 장면 기술 부분

비디오 객체에 대한 XMT-A에서의 장면 기술 부분을 그림 6에 나타내었다. 이 그림에서 살펴보면 그림 4에 나타난 이미지 객체에서와는 line 10 부분만 다를 수 있다. 즉, 이미지 객체에 대한 장면 기술에서는 <ImageTexture> 노드 엘리먼트를 사용하였지만, 비디오 객체에 대해서는 <MovieTexture> 노드 엘리먼트를 사용한다. 여기서, startTime은 비디오 객체의 디스플레이를 시작하는 시간을 나타내고, stopTime은 비디오 객체의 디스플레이를 끝내는 시간을 나타낸다. stopTime="-1"은 디스플레이를 무한히 계속한다는 것을 나타내고, 비디오 객체의 지속 시간이 지난 후에는 마지막 프레임을 계속 디스플레이한다

```

001     - <Transform2D DEF="Texture6" translation="-57 -90">
002     - <children>
003     - <Shape DEF="Video6">
004     - <geometry>
005     - <Bitrap scale="320 240" />
006     </geometry>
007     - <appearance>
008     - <Appearance>
009     - <texture>
010     <MovieTexture DEF="Texture6" url="obj9"
011     stopTime="-1" startTime="0" />
012     </texture>
013     </Appearance>
014     </appearance>
015     </Shape>
016     </children>
017     </Transform2D>
    
```

그림 6. 비디오 객체에 대한 장면 기술 부분

3.2. 객체 기술 부분

비디오 객체에 대한 객체 기술을 그림 7에 나타내었다. 이 그림에서 line 007과 line 012를 살펴보면, streamFormat=MPEG4-VideoFDIS, objectTypeIndication=0x22로 되어 있다. 이는 MPEG에서 제공하는 소프트웨어에서 MPEG-4 비디오를 지원하고 있지만 (streamFormat=MPEG4-Video, objectTypeIndication=0x20), 임의 모양을 가지는 비디오 객체에 대해서는 지원을 하지 않고 있기 때문에, 임의 모양을 가지는 비디오 객체까지 지원하는 MPEG-4 video 객체를 이와 같이 정의하여 사용하고 있다. 또한 line 014의 customData=320 240 binary는 임의 모양을 가지는 비디오 객체의 최대 크기가 320x240이고, 바이너리 알파맵을 가짐을 나타낸다.

4. 오디오 객체의 구성

4.1. 장면 기술 부분

오디오 객체에 대한 XMT-A에서의 장면 기술 부분을 그림 8에 나타내었다. 오디오 객체는 이미지나 비디오 객체와는 달리 시각적인 특성이 없기 때문에, <Shape> 노드 엘리먼트를 사용하지 않고, <Sound2D>를 이용하여 <AudioSource> 노드 엘리먼트에 객체에 대한 url, startTime, stopTime을 명시하고 있다.

```

001 - <ObjectDescriptor ObjectDescriptorID="od9">
002 - <Desc>
003 - <esDesc>
004 - <ES_Descriptor ES_ID="es9">
005 - <ScreenSource url="file:///sbs320x240_1560_151.miv">
006 <EncodingHints streamFormat="MPEG4-VideoH263" />
007 </ScreenSource>
008 </esDesc>
009 - <DecoderConfigDescriptor streamType="4">
010 bufferSizeB="78000"
011 objectTypeIndication="0x22">
012 - <DecoderSpecificInfo>
013 <UserDecoderSpecificInfo customData="320
014 240 Binary" />
015 </DecoderSpecificInfo>
016 </DecoderConfigDescriptor>
017 </Desc>
018 </ObjectDescriptor>
019 </esDesc>
020 - <SIConfigDescriptor ORResolution="1000"
021 AUSeqUnitLength="8"
022 timeStampLength="16"
023 useTimeStampsFlag="True"
024 packetSeqUnitLength="8"
025 timeStampResolution="1000"
026 useAccessUnitStartFlag="True"
027 useAccessUnitStartFlag="True"
028 useRandomAccessPointFlag="True" />
029 </SIConfigDescriptor>
030 </ES_Descriptor>
031 </Desc>
032 </ObjectDescriptor>
033 </ObjectDescriptor>
034 </OD>
    
```

그림 7. 비디오 객체에 대한 객체 기술 부분

```

001 - <ObjectDescriptor ObjectDescriptorID="od10">
002 - <Desc>
003 - <esDesc>
004 - <ES_Descriptor ES_ID="es10">
005 - <ScreenSource url="file:///weather1.723">
006 <EncodingHints streamAttr="6.3" streamFormat="G723" />
007 </ScreenSource>
008 </esDesc>
009 - <DecoderConfigDescriptor streamType="5">
010 bufferSizeB="9800"
011 objectTypeIndication="0xc1">
012 - <DecoderSpecificInfo>
013 <UserDecoderSpecificInfo customData="obsolete
014 string" />
015 </DecoderSpecificInfo>
016 </DecoderConfigDescriptor>
017 </Desc>
018 </ObjectDescriptor>
019 </esDesc>
020 - <SIConfigDescriptor ORResolution="1000"
021 AUSeqUnitLength="8"
022 timeStampLength="16"
023 useTimeStampsFlag="True"
024 packetSeqUnitLength="8"
025 timeStampResolution="1000"
026 useAccessUnitStartFlag="True"
027 useAccessUnitStartFlag="True"
028 useRandomAccessPointFlag="True" />
029 </SIConfigDescriptor>
030 </ES_Descriptor>
031 </Desc>
032 </ObjectDescriptor>
033 </ObjectDescriptor>
034 </OD>
    
```

그림 9. 오디오 객체에 대한 객체 기술 부분

```

001 - <transform2D DEF="Trans0" translation="-296 216">
002 - <children>
003 - <Sound2D DEF="Audio0">
004 - <source>
005 <AudioSource url="od10" startTime="-1" startTme="0" />
006 </source>
007 </Sound2D>
008 </children>
009 </transform2D>
    
```

그림 8. 오디오 객체에 대한 장면 기술 부분

5.1. 장면 기술 부분

텍스트 객체에 대한 XMT-A에서의 장면 기술 부분을 그림 10에 나타내었다. 텍스트 객체의 속성으로는 텍스트 스트링과 폰트 스타일 등이 있다. 이 예에서는 텍스트 객체를 나타내기 위하여 <Text> 노드 엘리먼트 및 이의 자식 노드로 <FontStyle>을 사용하고 있다. 텍스트 객체의 스트링은 <Text>의 attribute인 string=한국지도를 이용하여 나타내고 있으며, <FontStyle>을 이용하여 텍스트의 크기 (size), 스타일 (style), 글꼴 (family), 글자 간격 (spacing), align 방법 (justify, horizontal, leftToRight, topToBottom) 등을 지정한다.

4.2. 객체 기술 부분

오디오 객체에 대한 객체 기술 부분을 그림 9에 나타내었다. 이 그림에서 line 007을 살펴보면 streamFormat=MPEG4-VideoH263으로부터 G.723으로 부호화된 오디오 임을 알 수 있고, streamAttr=6.3은 부호화된 오디오 스트림의 비트율이 6.3 Kbps임을 나타낸다. 또한, line 012를 보면 objectTypeIndication=0xC1으로, objectTypeIndication을 user private으로 사용(표2 참조)한다.

5. 텍스트 객체의 구성

텍스트 객체의 경우에는 장면 기술을 통하여 모든 속성을 정의할 수 있고, 다른 미디어 객체가 필요 없기 때문에, 객체 기술 부분을 가지지 않는다.

```

001 - <Transform2D DEF="Trans10002" translation="169 158">
002 - <children>
003 - <Shape DEF="Text10002">
004 - <geometry>
005 - <text string="한글자">
006 - <fontStyle>
007 <FontStyle size="16.0" style="PLAIN"
008 family="SERIF" justify="BEGIN"
009 spacing="0.0" horizontal="TRUE"
010 leftRight="TRUE"
011 topBottom="TRUE" />
012 </fontStyle>
013 </text>
014 </geometry>
015 </Shape>
016 </children>
017 </Transform2D>
    
```

그림 10. 텍스트 객체에 대한 장면 기술 부분

IV. 결론

본 논문에서는 MPEG-4 콘텐츠를 텍스트추얼 포맷인 XMT를 이용하여 표현하는 방법에 대하여 설명하였다. XMT는 MPEG-4 Systems 규격에 정의되어 있는 MPEG-4 콘텐츠를 텍스트로 기술하기 위한 언어로서, 인터랙티브 멀티미디어 콘텐츠 저작 언어로 새롭게 부상하고 있다. 앞에서 기술한 예제들은 텍스트추얼 포맷인 XMT-A를 바이너리 포맷인 MP4로 변환하는 XMT-A to MP4 converter를 이용하여 mp4 파일로 변환한 뒤, MPEG에서 제공하는 reference software인 IM1-2D player로 display 할 수 있다.

참고 문헌


[1] ISO/IEC JTC1/SC29WG11, MPEG2001/M6871, Study of ISO/IEC 14496-1:2000/PDAM2, January 2001.
 [2] ISO/IEC 14772-1, The Visual Reality Modeling Language (VRML), 1997, <http://www.web3d.org/technicalinfo/specifications/vrml97/index.htm>.
 [3] ISO/IEC 14496-1, Coding of Audio-Visual Objects: Systems, October 1998.
 [4] ISO/IEC 14496-1, Coding of Audio-Visual Objects: Systems Amendment 1, December 1999.
 [5] Web3D Consortium, VRML 200x work in progress, <http://www.web3d.org/TaskGroups/x3d/specification>.
 [6] Extensible Markup Language (XML) 1.0, W3C recommendation, <http://www.w3.org/TR/REC-xml>, 10

February 1998.

[7] Namespaces in XML, <http://www.w3.org/TR/REC-xml-names/>, W3C Recommendation, 14 January 1999.
 [8] Synchronized Multimedia Integration Language(SMIL 2.0), <http://www.w3.org/TR/smil20/>, W3C Proposed Recommendation, 05 June 2001.

이 법 기(Bub-Ki Lee) 종신회원
 1992년 2월 : 경북대학교 전자공학과(공학사)
 1994년 2월 : 경북대학교 대학원 전자공학과(공학석사)
 1999년 8월 : 경북대학교 대학원 전자공학과(공학박사)
 2000년 4월 ~ 현재 : 기술신용보증기금 대전기술평가센터 <관심분야> : 영상신호처리 및 압축, 멀티미디어

정 원 식(Won-Sik Cheong) 정회원
 1992년 2월 : 경북대학교 전자공학과(공학사)
 1994년 2월 : 경북대학교 대학원 전자공학과(공학석사)
 2000년 2월 : 경북대학교 대학원 전자공학과(공학박사)
 2000년 5월 ~ 현재 : 한국전자통신연구원 방송미디어 연구부
 <관심분야> : 영상신호처리 및 압축, 멀티미디어

고 일 석(II-Seok Ko) 종신회원

 1989년 : 경북대학교 전자계산 전공 (공학사)
 1996년 : 경북대학교 컴퓨터공학과 (공학석사)
 M)USIU College of Business Administration(MBA)
 성균관대학교 경영대학원 앤소프전

약스쿨 수료
 연세대학교 컴퓨터산업시스템공학과 박사수료
 현재 : 충북과학대학 전자상거래과 교수

최 영 수(Young-Soo Choi) 정회원
 1991년 2월 : 경북대학교 전자공학과(공학사)
 1994년 2월 : 경북대학교 대학원 전자공학과(공학석사)
 2000년 2월 ~ 현재 : 충북대학교 대학원 정보통신공학과 박사과정

1995년 6월 ~ 현재 : 한국원자력연구원 양자광학팀

<관심분야> : 영상신호처리 및 압축, 멀티미디어

한 찬 호(Chan-Ho Han)

정회원

1990년 2월 : 경북대학교 전자공학(공학사)

1992년 2월 : 경북대학교 대학원 전자공학과 (공학석사)

2002년 2월 : 경북대학교 대학원 전자공학과 박사과정수료

1990년 1월 ~ 1997년 7월 : 현대전자(주) 미디어연구소

2000년 3월 ~ 현재 : 경운대학교 소프트웨어공학과

전임강사

<관심분야> : 영상신호처리 및 압축, 멀티미디어