

# 시스템 정보가 포함된 MOV 호환 MPEG-4 파일 포맷터의 설계 및 구현

전형국<sup>0</sup> 마평수  
한국전자통신연구원 인터넷정보가전연구부  
(hkjun, pmah)@etri.re.kr

## Design and Implementation of MPEG-4 File Formatter included MPEG-4 System Information, with MOV Compatibility

Hyung-Kook Jun<sup>0</sup> Pyeong-Soo Mah  
Dept. of Internet Appliance, Electronics and Telecommunications Research Institute

### 요 약

본 논문에서 제시하는 시스템은 MPEG-4로 부호화된 비디오/오디오에 대해서 MPEG-4 파일 포맷에 맞는 MPEG-4 파일 생성 포맷터이다. 본 논문에서 설계하고 구현된 파일 포맷터는 전처리 부분, 부호화기, 파일 포맷터로 구성된 저작도구의 모듈로서 존재한다. 따라서 본 논문에서는 저작도구에서 사용될 MPEG-4 파일 포맷터의 구조 및 기능에 대해서 설명하고, 포맷터의 세부 설계 사항을 기술한다.

본 논문에서 MPEG-4 파일 포맷터의 기능은 비디오/오디오 인코더로부터 출력되는 비디오/오디오 데이터를 MPEG-4 파일 포맷의 형식에 맞는 MPEG-4 파일을 생성하는 것이다. 생성된 파일은 비디오/오디오 각각에 대한 정보와 생성된 비디오/오디오를 처리할 디코더에 대한 정보를 제공한다. 표준 상의 MPEG-4 시스템에서의 동기화 정보를 파일 포맷상의 비디오/오디오 데이터에 제공하여, 스트리밍 서비스를 제공하는 서버쪽에서의 동기화 과정과 먹싱을 배제할 수 있게 한다. 비디오/오디오 인코더로부터 생성되는 다양한 종류의 비디오/오디오 데이터를 효율적으로 포맷팅하여 다양한 범위의 응용에서 사용될 수 있게 하며, 차기에 계위화 부호화를 효율적으로 적용시키기 위해 파일 포맷팅을 수행하게 된다.

### 1. 서 론

MPEG-4 파일 형식(Format)은 미디어의 상호교환, 관리, 편집, 화면 재생을 수월하게 하도록 유연하고 확장 가능한 형식으로 MPEG-4 미디어 데이터를 포함시키는 것이다. 화면 재생은 화면 재생 데이터가 포함되어 있는 시스템 상에서 될 수도 있고, 네트워크나 TransMux와 같은 다른 비트열 전달 체계를 통하여 될 수도 있다. 파일 형식은 특정 TransMux와는 독립적으로 동작하도록 고안되어야 하며, MPEG-4 파일 형식의 목적은 다음 기능을 제공한다.

- TransMux와 무관한 비트열의 전달 (TransMux Independence)
- 효과적으로 TransMux를 지원할 수 있는 파일 형식의 구현 (TransMux Support)
- 다양한 형태의 기초비트열(Elementary Stream)을 처리할 수 있는 기능 (Elementary Stream Management)
- 확장성 (Extensibility)
- 형식의 상호 교환 (Exchange Format)
- 임의접근 (Random Access)

MPEG-4 파일 형식은 다양한 분야에서 미디어의 상호 교환사용(Interchange), 데이터의 저장, 비트열 전송을 위한 준비, 시스템 내에서의 화면 재생 등 다양한 분야에서 사용 가능한데, 구현된 시스템에서는 비트열 전송을

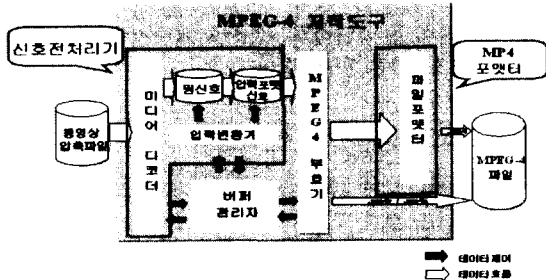
위한 준비 과정으로 사용된다.

본 논문에서 설계하고 구현한 파일 포맷터는 MPEG-4로 부호화된 비디오/오디오에 대해서 MPEG-4 파일 포맷에 맞는 파일을 생성하는 포맷터이다. 본 논문에서 설계하고 구현된 파일 포맷터는 전처리 부분, 부호화기, 파일 포맷터로 구성된 저작도구의 모듈로서 존재한다. 전처리 부분은 각각 다른 형식의 입력 파일에 대해서 하나의 공통 입력 데이터를 생성하는 부분이며, 부호화기는 비디오/오디오에 대해서 MPEG-4 부호화를 수행하는 부분이다. 따라서 본 논문에서는 저작도구에서 사용될 MPEG-4 파일 포맷터의 구조 및 기능에 대해서 설명하고, 포맷터의 세부 설계 사항을 기술한다.

생성된 파일은 비디오/오디오 각각에 대한 정보와 생성된 비디오/오디오를 처리할 디코더에 대한 정보를 제공한다. 표준 상의 MPEG-4 시스템에서의 동기화 정보를 파일 포맷상의 비디오/오디오 데이터에 제공하여, 스트리밍 서비스를 제공하는 서버쪽에서의 동기화 과정과 먹싱을 배제할 수 있게 한다. 비디오/오디오 인코더로부터 생성되는 다양한 종류의 비디오/오디오 데이터를 효율적으로 포맷팅하여 다양한 범위의 응용에서 사용될 수 있게 하며, 차기에 계위화 부호화를 효율적으로 적용시키기 위해 파일 포맷팅을 수행하게 된다.

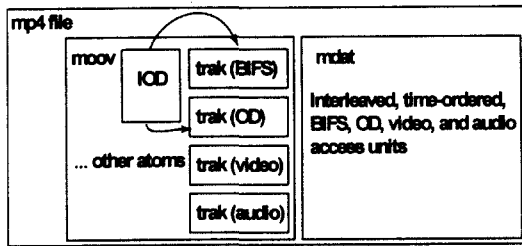
2. MPEG-4 파일 포맷터

MPEG-4 저작도구내의 파일 포맷터의 구성은 <그림 1> 과 같다. MPEG-4 저작도구내에서 파일 포맷터의 역할은 <그림 1>과 같이 MPEG-4 부호화기로부터 생성되는 비디오/오디오 데이터와 비디오/오디오 정보를 통해서 MPEG-4 파일을 생성한다. 이때 입력으로 제공되는 비디오/오디오 신호는 가변적이기 때문에 비디오/오디오 인코더에서 제공하는 Frame별 발생 비트수, Frame Rate, Bit Rate, Time Stamp등과 같은 정보를 바탕으로 비디오/오디오에 대한 파일 포맷팅을 수행하게 된다.



<그림 1> MPEG-4 저작도구 내에서의 파일 포맷터의 구성

아래의 <그림 2>는 MPEG-4 파일의 개략적인 파일 포맷의 구조를 보여준다.

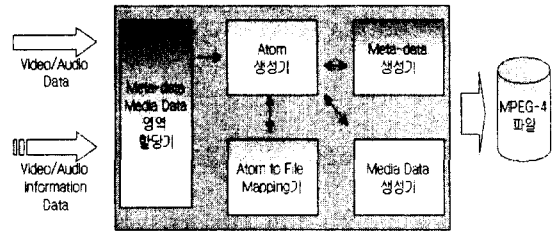


<그림 2> MPEG-4 파일 구조

그림에서 MPEG-4 파일의 구조는 크게 프레젠테이션과 관련되어 Meta-data 정보를 제공하는 moov와 실제 Media에 대한 데이터를 제공하는 mdat로 나뉘어진다. MPEG-4 포맷을 구성하는 단위는 Object로 여기서 Object는 moov와 mdat를 구성하는 최소 단위의 의미로 해석하면 된다. moov Object(Atom)는 앞에서 설명한 것 같이 프레젠테이션을 위한 모든 Meta-data 정보를 가지고 있어서, MPEG-4 표준을 위하여 IOD, OD, BIFS, 미디어 데이터에 대한 Meta-data 정보를 제공해 주게 된다. 이 Meta-data 정보를 통해서 실질적인 Media 데이터(mdat)에 대한 접근과 해석을 할 수 있게 된다.

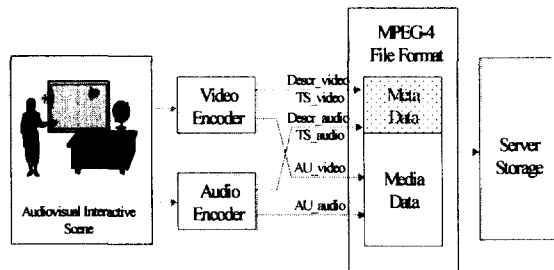
3. 파일포맷 생성기 및 재생기의 전체적인 동작

파일포맷 생성기는 <그림 1>과 같이 구성된다. 비디오와 오디오 부호화기를 통하여 부호화된 미디어 데이터(AU)들을 결합시켜서 파일포맷으로 만드는데, 이때 각 미디어에 대한 기술어(descriptor) 정보와 시간 정보인 TS(timestamp)가 함께 사용된다. 생성된 파일포맷

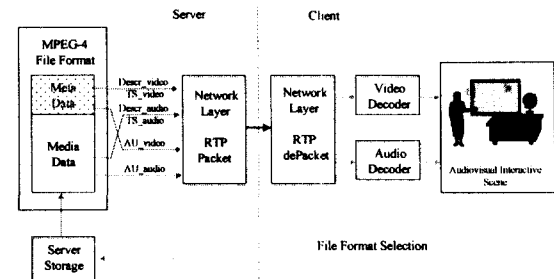


<그림 3> 파일포맷 생성기

데이터는 서버의 저장공간에 저장된다. <그림 5>와 같이 파일포맷 재생기는 클라이언트의 요구를 받아 서버의 저장공간에 있는 해당 파일포맷의 미디어 데이터에 있는 압축된 비디오와 오디오 데이터를 메타 데이터에 있는 시간정보와 기술어정보를 이용하여 각각의 AU단위로 분리하고 시간정보에 따라서 이 데이터를 네트워크 전송단으로 보내어 해당 RTP 패킷으로 만들어서 클라이언트쪽으로 전송하도록 한다.



<그림 4> 파일포맷 생성기의 흐름 구조도



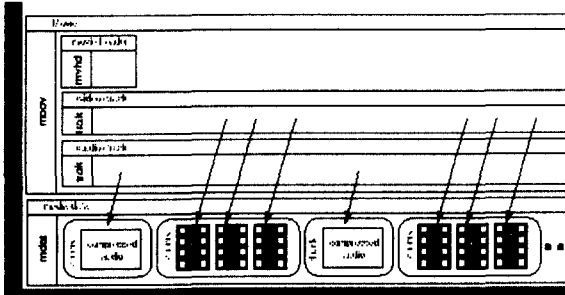
<그림 5> 파일포맷 재생기의 구조

4. 화면 재생 구조

4.1 파일 구조

하나의 화면 재생이 몇 개의 파일에 포함될 수도 있다. 하나의 파일은 전체 화면 재생을 위한 메타 데이터를 포함한다. 또한 모든 미디어 데이터를 포함할 수도 있다. 하나의 MPEG4 파일은 미디어 트랙에 미디어 데이터를 반드시 포함해야 한다. 파일 내에서 미디어 데이터의 배치는 <그림 6>과 같이 된다. 각 미디어 데이터는 화면에 디스플레이되는 순서로 차례대로 저장된다. 그리고 각 미디어 데이터는 해당 미디어 부호화기에서 부호화된 비트열이 AU단위로 저장되어 있다.

<그림 6>은 또한 비디오와 오디오가 모두 포함되는 경우에 파일의 블록도의 한 예를 나타낸다. 트랙과 미디어 데이터 사이의 관계는 타이밍 정보를 포함하는 Sample Table에 의해 지정된다.



<그림 6> 비디오와 오디오가 모두 포함된 파일의 구조

5. 파일포맷의 개략적인 전체 구조

파일 구조의 각각의 형태를 구분하기 위해 사용된 지정자는 <표 1>과 같다. 모든 지정자는 4바이트의 문자로 구성된다. 이 지정자들 중의 일부는 다른 지정자의 하위 계층을 구성하기도 한다.

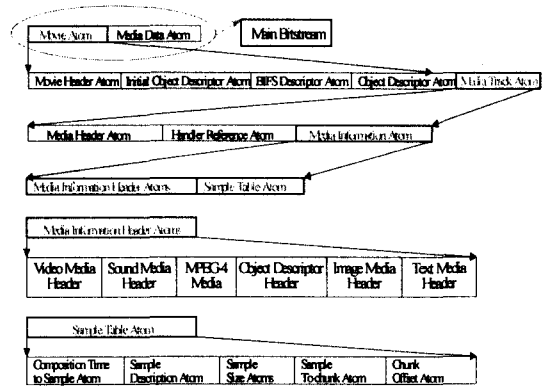
<표 1> 사용된 지정자와 의미

지정자	의미
'moov'	Meta Data Atom
'mdat'	Media Data Atom
'mvhd'	Movie Header Atom
'iods'	Object Descriptor Atom
'trak'	Media Track Atom
'mhdh'	Media Header Atom
'hdlr'	Handler Reference Atom
'minf'	Media Information Atom
'vmhd'	Video Media Header Atom
'smhd'	Sound Media Header Atom
'hmhd'	Hint Media Header Atom
'stbl'	Sample Table Atom
'stts'	Composition Time To Sample Atom
'std'	Sample Description Atom
'stsz'	Sample Size Atoms
'stsc'	Sample To Chunk Atom
'stco'	Chunk Offset Atom

- Movie Atom  
실제 재현되는 미디어 데이터에 관한 meta 데이터, 즉 헤더 정보가 정의
- Movie Header Atom  
전체 화면 구성에 관한 미디어의 정보를 정의
- Object Descriptor Atom  
Scene stream을 위한 ES\_Descriptor와 Object Descriptor을 위한 ES\_Descriptor를 포함
- Media Track Atom  
비디오, 오디오, 텍스트, 정지영상 등 각각의 미디어에 관한 정보를 정의
- Handler Reference Atom  
비트열이 무엇에 의해 handle되는지를 정의한다. 즉, handler의 type을 정의한다.
- Media Information Atom

비트열내에 있는 미디어의 특성 정보를 선언하는 필드로서, Media Information Header Atom, Sample Table Atom등으로 구성

- Media Information Header Atom  
Media Handler Type에 대응하는 각 track type에 관한 media information header를 정의하는 부분
- Sample Table Atom  
한 트랙에 있는 샘플(sample)의 개수와 기간, 샘플의 위치, 형태등을 정의(Time to Sample Atom, Sample Description Atom, Sample Size Atom, Sample to Chunk Atom, Chunk Offset Atom)
- Media Data Atom  
비디오, 오디오의 ES가 AU(access unit)단위로 저장. 본 파일 포맷터에서 생성한 개략적인 파일 포맷의 구조는 <그림 7>과 같다.



<그림 7> 개략적인 파일 형식

6. 결론 및 향후 연구 과제

본 논문에서 제안하는 파일 포맷터는 MPEG-4 파일 포맷 스펙에 의해 구현되었으며, MPEG-4 시스템의 기능들을 배제하기 위하여 파일 포맷 구성시에 시스템에서 제공해야 될 정보들을 제공해 주고 있다. 따라서 본 논문에서 구현된 파일 포맷터는 포맷팅시 동기화 정보와 MUXING을 수행하며, MPEG-4 표준과 유사한 MOV 포맷을 수용하기 위한 구조를 가지도록 만들어 졌다. 향후 연구과제로 본 파일 포맷터는 사용자의 요구에 따라 동기화 정보와 MUXING 단위를 변화 시킬 수 있는 포맷터를 설계하며 scalable 포맷의 지원과 클라이언트 서버간의 MPEG-4 미디어 스트림을 위한 VCR 연산 기능들이 포함될 것이다.

7.참고 문헌

[1] ISO/IEC FCD 14496-1 " Systems", 1999-12-15.  
 [2] ISO./IEC 14496-1/FDAM-1 " MPEG-4 version 2 Intermedia Format-MP4", 1999.11  
 [3].ISO/IEC JTC1/SC29/WG11 N3536, MPEG-4 Overview, July 2000  
 [4].http://www.microsoft.com/windows/windows-media/en/default.asp  
 [5].http://www.realnetworks.com