

MPEG-21 파일 포맷 - 미디어리소스 어드레싱 방법 및 보안

조용주, 임영권, 송문섭, 김형중
한국전자통신연구원, 넷앤티브이, 픽스트리, 강원대학교

MPEG-21 File Format - Media Resource addressing technique and security

Yongju Cho, Youngkwon Lim, Moonsep Song, Hyung-Joong Kim
ETRI, Net&TV, Pixtree, Kangwon National University

Abstract - 본 논문에서는 MPEG-21 파일 포맷의 구조, 미디어리소스를 어드레싱 하는 방법, 그리고 파일의 보안 방법을 제안하였다. 제안한 파일 포맷은 MPEG-4 터미널에서도 상호 운용 가능케 하는 구조 (backward compatibility with MPEG-4)이며, XML로 구성된 디지털아이템에서 정의된 미디어리소스를 효과적으로 참조하는 방법 등이다. 또한, 파일의 보안을 위한 기본 틀을 제시 하였다. 본 논문에서 정의된 파일 포맷은 디지털 방송, 무선통신, 인터넷 환경에서의 다양한 형태의 콘텐츠들을 효과적이고 체계적으로 조합할 수 있는 기능 및 효과적인 전달 기능을 제공 하며, 향후 XML기반 메타데이터와 미디어리소스를 포함하는 멀티미디어 콘텐츠의 파일 포맷에 대한 참고 모델로서 사용될 수 있다.

1. 서 론

MPEG-21 표준은 다양한 네트워크 환경에서 호환 가능한 멀티미디어 리소스의 사용을 증대하기 위한 프레임워크를 정의하자는 목표로 2000년부터 표준화가 시작되었으며, 현재 MPEG-21 파일 포맷은 part 9으로 정의되어 상기 목적을 충족할 수 있는 새로운 구조의 멀티미디어 파일 포맷을 정의 하기 위해 활발하게 진행 중에 있다.

본 논문에서는 MPEG-21 디지털아이템 선언, 참조(reference)된 정지영상, mp4등과 같은 미디어리소스들이 하나의 파일 내에 포함될 수 있는 세부적 방법을 설명한다. 특히, XML기반 텍스트로 제작된 디지털아이템을 멀티미디어 파일 포맷화 하기위해 디지털아이템 선언에서 미디어리소스를 참조하는 방법, 그리고 멀티미디어 파일을 보호할 수 있는 방법에 대하여 자세히 설명하고 있다. 제안한 멀티미디어 파일 포맷의 구조 및 기능은 추후 메타데이터와 미디어리소스의 조합으로 구성된 멀티미디어 콘텐츠의 전달을 위하여 하나의 파일로 제작할 때 참고 모델로서 유용하게 사용될 수 있다.

본 논문은 다음과 같이 구성되어 있다. 2장에서는 MPEG-21 파일 포맷 구조 및 기능에 관하여 자세히 설명하고, 3장에서는 결론을 맺는다.

2. MPEG-21 파일 포맷

2.1 MPEG-21 콘텐츠 (디지털아이템)

디지털아이템은 MPEG-21 멀티미디어 프레임워크 내에서 표준화된 표현, 식별, 그리고 메타데이터를 지니는 구조화된 객체이다. 이는 효과적인 멀티미디어 제공 서비스를 위해 사용자로 하여금 콘텐츠에 대한 탐색을 가능케 하고, 저작권자의 권리를 이용자에게 알리며, 또한 콘텐츠의 이용에 관련된 정보 전달을 위한 메타데이터를 제공한다.

디지털아이템 선언은 디지털아이템의 구조체를 정의하는 부분으로 디지털아이템 서술 언어(Digital Item Declaration Language) 을 이용한다. 이는 디지털아이템에 관련된 콘텐츠 및 메타데이터의 상호 관련성을 정의하고 있는 언어로서 XML Schema로 정의되어 있어 XML언어가 지니는 일반적인 특징인 유연성 및 확장성을 지닌다. 표1은 16개의 요소(element)들의 조합으로 제작된 디지털아이템의 일례를 보여주고 있다.

2.2 MPEG-21 파일 포맷 구조

MPEG-21 파일 포맷은 메타데이터와 멀티미디어 리소스들(예-동영상, 정지영상)을 효과적이고 체계적으로 패키징하기 위한 구조를 갖는다[1]. 또한, MPEG-21 파일 포맷은 MPEG-4 파일과 상호 양립해야 하는 요구사항을 충족 시키는 구조이다[2]. 상기 목적을 위하여 MPEG-21 파일 포맷은 meta-data, mdat, moov 박스로 구성된다. 상기 파일의 구조는 그림 1에서 잘 나타내고 있다.

2.2.1 meta-data 박스

meta-data 박스는 handler box, DataInformation box, ItemLocation box, DID box로 구성된다[3]. 추후 특별한 목적을 위한 box추가도 가능하다. handler box는 meta-data box의 타입을 정의하는 정보를 포

함하며, DataInformation box는 외부에 존재하는 미디어리소스의 참조(reference)정보를 담고 있는 테이블을 포함한다. ItemLocation box는 파일 내 또는 외부에 존재하는 미디어리소스의 위치정보를 offset과 length정보를 이용하여 나타낸다. 이 논문에서 제시한 미디어리소스의 어드레싱 방법이 표현되는 곳이기도 하다. DID box는 XML기반의 디지털아이템을 포함한다.

2.2.2 Mdat box

Mdat box는 미디어리소스들을 포함한다. 미디어리소스는 오디오, 비디오, 정지영상, MPEG-4 파일(mp4)등 터미널에서 소비 가능한 콘텐츠를 의미한다. Mdat내에는 1개 이상의 mp4파일을 포함할 수 있다는 것이 특징이며, 각 미디어리소스는 meta-data box 내 ItemLocation box로부터 참조(address) 받게 된다. MPEG-21 파일(mp21)은 meta-data box와 Mdat으로 구성되며, MPEG-4 파일(mp4)은 moov와 Mdat box의 조합으로 구성된다. 이는 MPEG-21 파일 포맷 요구 사항인 MPEG-4 backward compatibility를 충족시켜주기 위함이다. mp4인 경우, Mdat내 미디어리소스는 moov내 trak정보를 통해 참조 된다(그림 1, 2).

2.2.3 Moov 박스

Moov는 Mdat과 함께 mp4를 구성하는 구조체로서, moov의 상세한 구조 및 기능들은 MPEG-4 systems(14496-1)에 자세히 설명되어 있다. 참고로 moov는 trak들의 집합이라 간주할 수 있고, 각 trak은 미디어리소스의 정보를 담고 있는 "mdia"를 포함하고 있다. 즉, trak정보를 이용하여 Mdat내의 미디어리소스를 어드레싱 할 수 있는 것이다. 그림 2는 MPEG-4 파일 포맷의 구조를 나타내고 있다.

2.3 미디어리소스 어드레싱 방법

MPEG-21 파일 포맷에서 핵심이 되는 기술로서 메타데이터에서 정의한 미디어리소스를 어드레싱하기 위한 방법을 의미한다[4]. 미디어리소스는 파일 내 또는 외부에 존재가능하며, 본 절에서는 이 두 가지 경우를 충족시킬 수 있는 방법을 자세히 기술한다.

2.3.1. 파일 내부 미디어리소스 어드레싱

미디어리소스가 파일 내부에 포함될 경우 meta-data box에서 이 미디어리소스를 어드레싱하기 위하여 기본적으로 파일의 offset값을 사용한다. 또한 Mdat box내 포함된 mp4 파일의 특정 미디어리소스(Elementary Stream(ES))를 어드레싱하기 위하여

moov의 정보 또는 moov 내 trak정보를 이용한다. 다음은 미디어리소스를 어드레싱방법의 일례를 보여주고 있다.

1. 1000번지에 위치한 미디어리소스(mp4)를 어드레싱할 경우

```
<Resource target="#offset=1000" mimeType="video/mp4" >
```

2. mp4 file내 첫 번째 트랙에서 나타내는 ES(Elementary stream-예: video or audio)를 참조할 때

```
<Resource target="#offset=1000/moov/trak[1]" mimeType="video/mp4" >
```

3. 2와 같은 기능("/" 추가로 moov를 대체함)

```
<Resource target="#offset=1000//trak[1]" mimeType="video/mp4" >
```

4. mp4 file내 첫 번째 트랙에 존재하는 mdia에서 나타내는 ES를 참조할 때

```
<Resource target="#offset=1000/moov/trak[1]/mdia" mimeType="video/mp4" >
```

5. 4와 같은 기능("/" 추가로 moov를 대체함)

```
<Resource target="#offset=1000//trak[1]/mdia" mimeType="video/mp4" >
```

6. mp4 file 내 특정 ODID(Object Descriptor ID)와 ESID(Elementary Stream ID)값을 가진 Elementary stream을 참조할 때

```
<Resource ref="#offset=1000:odid=0x10&esid=0x11">
```

2.3.2. 파일 외부 미디어리소스 어드레싱

파일 내부 미디어리소스 어드레싱 방법과 기능적으로 매우 유사하며, URI를 이용하여 외부에 위치한 미디어리소를 어드레싱 하는 것이 차이점이다.

7. www.etri.re.kr에 존재하는 mp4 file(test.mp4)을 참조할 때

```
<Resource ref="http://www.etri.re.kr/test.mp4" mimeType="video/mp4">
```

8. mp4 file내 첫 번째 트랙에서 나타내는 ES(Elementary stream-예: video or audio)를 외부 참조할 때

```
<Resource target=" http://www.etri.re.kr/test.mp4:moov/trak[1]" mimeType="video/mp4" >
```

9. 8와 같은 기능("/" 추가로 moov를 대체함)

```
<Resource target=" http://www.etri.re.kr/test.mp4:trak[1]" mimeType="video/mp4" >
```

10. mp4 file내 첫 번째 트랙에 존재하는 mdia에서 나타내는 ES를 외부 참조할 때

```
<Resource target=" http://www.etri.re.kr/test.mp4:moov/trak[1]/mdia" mimeType="video/mp4" >
```

11. 10과 같은 기능("/" 추가로 moov를 대체함)

```
< R e s o u r c e
```

target="http://www.etri.re.kr/test.mp4:/trak[1]/mdia" mimeType="video/mp4" >

12. mp4 file(test.mp4)내 odid=0x10 과 esid=0x11 값을 가진 미디어리소스를 외부 참조할 때

```
< Resource URI create
target="http://www.etri.re.kr/test.mp4:odid=0x10&esid=0x11"
mimeType="video/mp4" >
```

13. 외부에 존재하는 mp21 file(test.mp21)내 offset=1000에 위치한 mp4 file를 구성하는 여러 ES들 중 odid=0x10 과 esid=0x11 값을 가진 미디어리소스를 외부 참조할 때

```
<Resource ref="
http://www.etri.re.kr/test.mp21#offset=1000:odid=0x10&esid=0x11"
>
```

2.4 파일 보안

Security box는 mp21 파일 내 메타데이터의 보호를 목적으로 제안한 box이다. 이 box 는 DID(다른 메타데이터 포함)내 특정한 부분을 요구하는 보호방법을 통해 보호할 수 있는 기능을 제공한다. 보호 방법에는 digital signature, encryption, watermark, CRC, Hash등이 있을 수 있다[5].

2.4.1 Syntax

표 2에서 상기 security box의 선택스를 나타내고 있다.

2.4.2 Semantics

- version은 이 box의 버전을 나타내는 integer이다
- entry_count는 entry(디지털아이템 내 보호할 element)의 수를 나타내는 integer이다
- element_ID는 디지털아이템 내 보호할 element의 ID값을 나타내는 integer이다
- security_type은 보호 방법을 나타내는 integer이다

예: Digital Signature, CRC, hash

- PayloadSize는 페이로드의 길이를 나타내는 integer이다
- Payload는 디지털아이템 내 선택된 element정보가 선택된 보호 방법을 통해 나온 결과 값이다
예: Signature value, hash value, CRC

3. 결 론

본 논문에서는 MPEG-21 디지털아이템의 전달을 목적으로 하는 멀티미디어 파일의 구조, 미디어리소스를 어드레싱 하는 방법, 그리고 파일을 보호할 수

있는 방법에 대해서 제안하였다.

제안된 멀티미디어 파일 포맷은 XML기반 디지털아이템선언과 미디어리소스를 하나의 파일 내에 체계적으로 구성함으로써 파일 내/외에 존재하는 미디어리소스의 random access를 가능하게 하며, 전달의 목적으로 사용하기에 적합하다는 것을 입증할 수 있었다. 향후 본 논문에서 제안한 멀티미디어 파일 포맷은 MPEG-21 File Format 표준화 방향에 맞추어 수정 보완된 후 표준 안으로 제안될 것이다.

감사의글

본 논문은 정보통신부 지원의 "MPEG-21 기반 방송. 통신 융합 시스템 기술 개발"과제의 일환으로 작성되었으며 관련 담당자에게 감사의 글을 전합니다.

(참 고 문 헌)

- [1] Proposed MPEG-21 Container File Format ISO/IEC 21000-9 1.0, ISO/IEC JTC1/SC29/WG11/N5095, July 2002, Klagenfurt
- [2] Draft Requirements for Digital Item Processing for Applications and MP21 File Format, ISO/IEC JTC1/SC29/WG11/N4802, May 2002, Fairfax
- [3] Storage of untimed meta-data in ISO media files, ISO/IEC JTC1/SC29/WG11/W5658, March 2003, Pattay
- [4] Yongju Cho, Youngkwon Lim, Hyungjoong Kim, "Proposed MPEG-21 File Format," ISO/IEC JTC1/SC29/WG11/M9486, March 2003.
- [5] 조용주, 홍진우, 김형중, 임영권, 김진웅, "MPEG-21 기반 멀티미디어 파일 포맷", 대한전자공학회 하계학술대회, 2003.

표1. 디지털아이템의 예

```
<?xml version="1.0" encoding="UTF-8"?>
<DIDL xmlns="urn:mpeg:mpeg21:2002:01-DIDL-NS"
xmlns:mpeg7="urn:mpeg:mpeg7:schema:2001"
xmlns:RDF="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:dc="http://purl.org/dc/elements/1.1/">
  <Item>
    <Descriptor>
      <Statement mimeType="text/xml">
        <mpeg7:Mpeg7>
          ...
        </mpeg7:Mpeg7>
      </Statement>
    </Descriptor>
    <Descriptor>
      <Statement mimeType="text/xml">
        <RDF:Description>
          <dc:title>When the Thistle Blooms</dc:title>
          <dc:creator>Always Red</dc:creator>
          <dc:publisher>PDQ Records</dc:publisher>
        </RDF:Description>
      </Statement>
    </Descriptor>
  </Component>
  <Resource ref="rtsp://telemedia1:/v11.mp4"
mimeType="audio/mp4a-latm"/>
  <Resource ref="urn:doi:10.1000-1" mimeType="audio/mp4a-latm"/>
```

```

</Component>
</Item>
</DIDL>
    
```

표 2. 보안 박스의 선택스

```

aligned(8) class secBox extend extends FullBox('sec', version
= 0, 0) { unsigned int(32) entry_count
    for(i=1; i<= entry_count; i++){
        unsigned int(32) element_ID;
        unsigned int(128) security_type;
        unsigned int(32) PayloadSize;
        unsigned int(8) Payload[];
    }
}
    
```

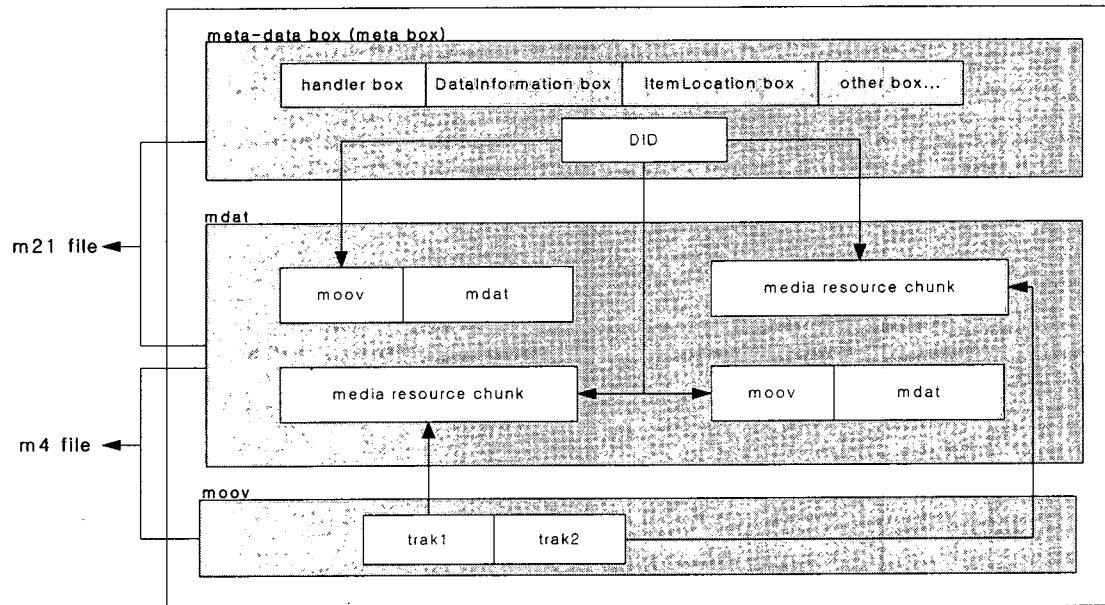


그림 1. MPEG-21 파일 포맷 구조도

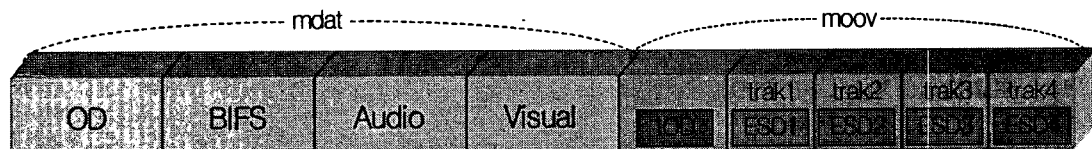


그림 2. MPEG-4 파일 포맷 구조도