

# 모바일 환경에서의 이진 부호화된 XML 문서를 위한 복호기 설계 및 구현

유정수 남종호  
서강대학교 컴퓨터공학부  
yjs@mlneptune.sogang.ac.kr, jhnang@sogang.ac.kr

## Design and Implementation of Decoder for Binary Encoded XML Document on Mobile Environment

Jeongsoo Yu Jongho Nang  
Department of Computer Science and Engineering, The Sogang University of Korea

### 요 약

최근 멀티미디어 응용에서 XML은 메타데이터를 표현하기 위하여 그 사용량이 증가하고 있는 추세이며, 그 예로는 MPEG-7 MDS, MPEG-21 등이 있다. 이러한 XML형식의 메타데이터의 크기를 줄이기 위하여 MPEG-7에서 정의한 포맷이 Binary Format for Metadata(BiM)이다. BiM은 스트리밍을 고려하여 단편화를 통한 점진적인 전송이 가능하며, 압축률이 높고 복호화에 따르는 오버헤드가 작다는 장점을 제공한다. 본 논문에서는 BiM에 대한 간단한 소개와 모바일 환경에 적합하도록 구현한 BiM 복호기의 설계 및 구현에 대하여 소개한다. 구현한 복호기는 모바일 환경의 다양한 플랫폼에서의 XML 파싱 모듈과 DOM 트리 모듈에 대한 종속성을 제거하기 위하여, 스키마 파일의 파싱을 데스크탑에서 수행하고 이에 대한 파싱 결과를 자체 정의한 Syntax File 형식으로 복호기에 전송할 수 있도록 하였다. 또, BiM형태로 전송되는 ESG 메타데이터와 MPEG-4 LAsER 스트림에 대하여 Windows, Windows CE, embeded Linux 환경에서 정상적으로 복호화 하는 것을 확인하였다.

### 1. 서 론

최근 XML은 멀티미디어 응용에서 메타데이터를 표현하기 위한 목적으로 사용량이 증가하고 있는 추세이며, 이에 대한 예로는 MPEG-7 Multimedia Description Scheme[1]과 MPEG-21 Usage Environment Description[2] 등을 들 수 있다. 그러나 XML은 고성능을 요구하거나 스트리밍이 가능한 또는 자원의 제약을 받는 환경을 목적으로 설계되지 않았으며, 따라서 모바일 환경 등에서의 멀티미디어의 요구가 증가함에 따라 이러한 XML 문서의 효율적인 전송방법이 필요하게 되었다. 이에 따라 MPEG은 높은 압축률과 스트리밍이 가능함을 제공하는 Binary Format for Metadata (BiM)을 MPEG-7 Part 1[3]에서 정의하였다. BiM은 MPEG-7의 멀티미디어 메타데이터를 부호화 하기 위한 목적으로 설계되었지만 스키마 종속적인 특성으로 인하여 특정 XML에 제한받지 않으며, 바이너리 상태에서 전체 XML다큐먼트의 서브트리에 대하여 전송 및 스킵이 가능하여 스트리밍에 유리하다는 장점이 있다. 또, 일반 텍스트 압축 방식에 비하여 BiM은 더 높은 압축률을 제공하며[4], 텍스트 형태의 XML문서를 파

싱하는것에 비하여 적은 리소스를 사용하며 복호기능을 수행할 수있다. 따라서 BiM은 XML문서의 스트리밍을 주 목적으로 하지만, 모바일 환경에서는 스트리밍뿐만 아니라 일반적인 XML처리에서도 텍스트 형식에 비하여 더 유리하다고 볼 수있다. 즉, BiM복호기는 모바일 환경에서, 많은 리소스를 요구하는 텍스트 형식의 XML 문서 처리에 대한 대안으로 사용 가능할 것이다.

본 연구에서는 이러한 BiM의 특성을 고려하여 모바일 환경에서 동작하는 복호기를 설계 및 구현하였다. 구현된 복호기는, 먼저 모바일 환경의 다양한 플랫폼을 고려하여, 포팅 계층을 두어 여러 플랫폼상에 포팅이 용이한 구조로 설계되었다. 그리고, BiM 복호기는 스키마를 기반으로 복호 기능을 수행하기 때문에 초기화 과정에서 스키마 파일을 컴파일해야 하지만, 스키마 파일이 텍스트 형태의 XML파일로 존재하는 경우가 일반적이기 때문에, 텍스트 XML 처리를 대신하기 위한 BiM 복호기가 텍스트 XML 파서를 필요로 한다는 문제점이 있다. 본 구현에서는 이러한 문제점을 해결하기 위하여 스키마 파일의 파싱을 데스크탑 환경에서 미리 수행한 후 본 연구에서

정의한 Syntax File Format의 형태로 단말 환경으로 전송하여 복호기가 사용할 수 있도록 설계하였다. 따라서, 본 연구에서 구현한 BiM복호기를 사용할 경우 열악한 모바일 환경에서 XML파서를 필요로 하지 않으며 XML처리가 가능하다는 효과가 있다.

2. 스키마에 기반한 XML 문서의 이진 부호화

BiM은 스키마를 기반으로 XML문서를 부호화 및 복호화 하도록 정의되어있다. 부호화된 바이너리 스트림은 XML 트리의 각 자식 엘리먼트(element)를 선택할 수 있도록 하며, 이러한 개별 엘리먼트는 스키마에 따라 구성된 자신의 타입에 대한 코덱(Type codec)을 통해 XML 트리로 복호화 된다. 즉, BiM 복호기는 바이너리 스트림에 의해 전이되는 유한 오토마타 형태로 구성된다.

BiM스트림은 크게 커멘드(command)와 컨텍스트(context), 페이로드(payload)로 이루어 진다. 커멘드는 복호화된 XML 서브트리를 전체트리에 적용시킬 오퍼레이션에 대한 정보로서, Add, Replace, Remove, Delete, Reset기능을 제공한다. 컨텍스트는 커멘드가 수행될 위치를 표현하며, 루트 노드로부터의 절대 경로 또는 이전 컨텍스트로부터의 상대 경로를 제공한다. 컨텍스트는 Tree Branch Code(TBC)라 불리는 로컬 네비게이션 정보들의 시퀀스로 구성되어있으며, 복호기는 스키마에 따 fms Complex Type의 형태를 알고 있으므로, 이러한 TBC들을 해석하여 XML 트리에서 각 노드의 자식노드를 선택하며 경로를 구할 수 있게 된다. <그림 1>과 표 1은 이러한 컨텍스트의 TBC에 대한 간단한 예제이다.

```
<complexType name="Audio">
  <complexContent>
    <element name="title", type="string"/>
    <attribute name="genre" type="GenreType"/>
    <attribute name="creator" type="PersonType"/>
  </complexContent>
</complexType>
```

<그림 1> Complex Type 예제

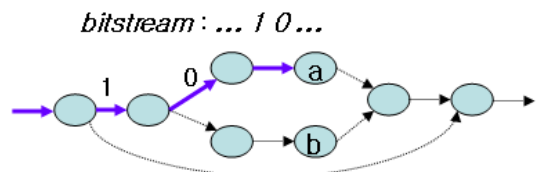
| TBC table |             |                 |                     |
|-----------|-------------|-----------------|---------------------|
| SBC       | Type Code   | Position Code   |                     |
| 000       | -           | -               | reference to parent |
| 001       | [Type Code] | [Position Code] | title               |
| 010       | [Type Code] | [Position Code] | genre               |
| 011       | -           | -               | creator             |
| 111       | -           | -               | Termination Code    |

표 1 TBC Table 예제

페이로드는 이진 부호화 된 XML 서브트리의 엘리먼트들을 이진 부호화 한 형태로 포함한다. 스키마에 정의된 XML 콤플렉스 타입(complex type)의 각 엘리먼트는 바이너리 스트림에 의한 유한 오토마타의 전이에 의해 선택되어지며, 선택된 엘리먼트는 자신의 타입에 따라 복호화 된다. <그림2>와 <그림 3>은 오토마타를 통한 페이로드 복호화를 간단히 나타내는 예제로서, 첫 번째 비트가 1이므로 optional group인 choice가 디코딩되며, 다음 비트가 0이므로 choice 노드의 자식 엘리먼트 중 A가 선택되는 것을 보여준다. 그다음, 오토마타의 a 상태에서 타입 a에 대한 디코딩이 수행되게 된다. 이 때 타입 a가 심플 타입(simple type) 이라면 해당 타입 코덱이 타입 a에 대한 복호화를 수행하며, 타입 a가 콤플렉스 타입이라면, 타입 a에 대한 오토마타에서 새로운 전이를 통한 복호화가 수행되게 된다. 정리하면, 페이로드는 XML 트리의 서브 요소들에 대하여 재귀적인 형태로 이진 부호화 되어있으며 스키마에 기반한 인스턴스는 유한 오토마타의 전이에 따라 결정되게 되고, 종단 노드는 해당 타입에 대한 코덱을 통하여 복호화 된다.

```
<choice minOccure="0">
  <element name="A", type="a"/>
  <element name="B" type="b"/>
</choice>
```

<그림 2> Complex type 예제



<그림 3> 페이로드 복호화 예제

그밖에도 BiM은 스키마의 버전에 따른 호환성을 제공한다. XML 스키마는 다형성(polymorphism) 또는 대체 방식(substitution)을 통한 변경이 가능하다. BiM은 엘리먼트의 부호화를 스키마의 버전을 기준으로 구분함으로써, 복호기는 알지 못하는 버전의 스키마에 정의된 엘리먼트를 스킵할 수 있다.

3. 모바일 환경을 위한 BiM 복호기 설계

본 연구에서는 모바일 환경에서 동작하는 BiM 복호기를 설계 및 구현하였다. <그림 4>는 일반적인 데스크탑

환경에서의 BiM 복호과정에 대한 설명이다. <그림 4>를 보면, BiM 복호기는 2장에서 설명한 바와 같이 스키마를 기반으로 복호를 수행하므로 텍스트 형식의 XML 문서인 스키마 파일을 파싱하기 위한 XML 파서를 필요로 한다. 본 구현에서는 복호기가 스키마로부터 직접 초기화 되는 구조 대신, <그림 5>와 같이 Syntax File(SF)을 통해 스키마 정보를 얻는 방식을 사용한다. SF는 스키마 파일이 파싱된 후 형성된 DOM 트리[5]를 운행한 결과를 복호기로 전송하기 위하여 본 구현에서 정의한 포맷이다. SF 파서는 SF를 읽으면서 복호기에 스키마 정보를 제공하며, 복호기는 마치 스키마를 파싱한 XML 파서의 DOM 트리를 직접 운행하면서 스키마 정보를 입력받는 것과 같은 효과를 얻을 수 있게 된다. 다음은 SF의 생성 방식에 대하여 간략히 설명한다.

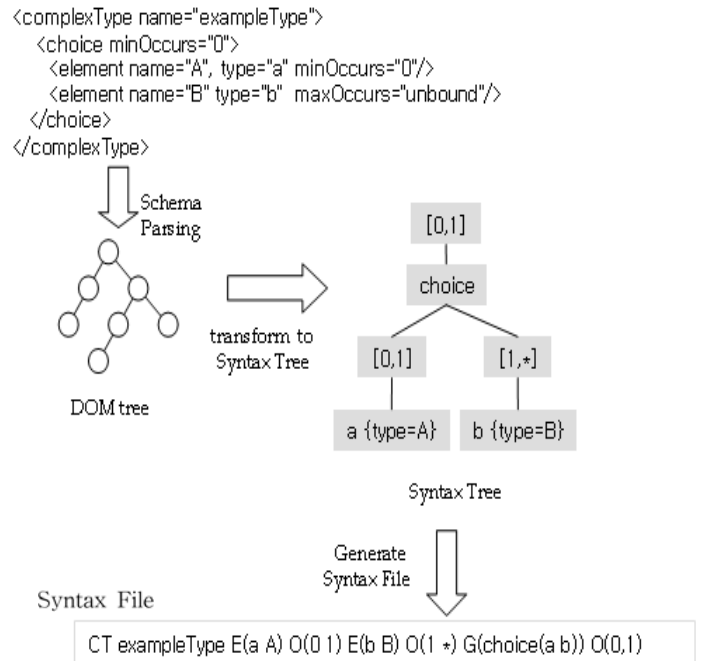
a. Complex Type에 대한 SF 생성

XML파서에 의해 파싱된 스키마 파일의 타입들은 DOM 트리 형태로 저장되고, 그중 Complex type은 선택스 트리(Syntax Tree)[3]형태로 변형 된다. 선택스 트리는 BiM 복호를 위한 유한 오토마타를 생성하기 위한 구조로써 Occurrence 노드와 Group 노드, Element 노드로 구성된다.

유한 오토마타는 선택스 트리를 후위 순서로 운행하며 각 노드의 종류에 따라 정해진 방식대로 오토마타를 형성하게 된다. 콤플렉스 타입에 대한 SF 생성의 기본 아이디어는 이러한 선택스 트리의 후위 우선 순위를 텍스트 형태로 기록하여, XML파서가 없는 플랫폼에서 SF를 사용하여 선택스 트리를 후위 운행하는 효과를 얻는 것이다. <그림 6>은 이와 같은 콤플렉스 타입에 대한 SF의 생성 과정에 대하여 간단한 예를 통하여 나타내고 있다.

b. Simple Type에 대한 SF 생성

심플 타입은 기본 타입(Primitive Type)들에 대한 상속



<그림 6> Complex Type에 대한 Syntax File 생성 과정

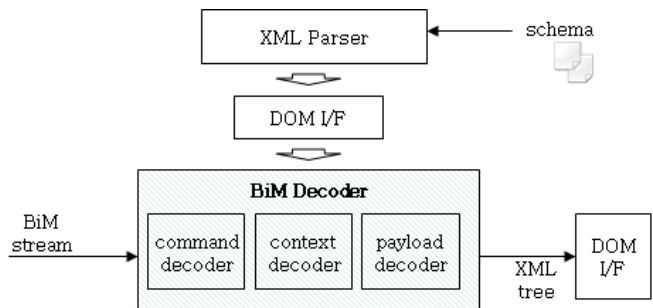
```

Schema :
<simpleType name="rareType">
  <restriction base="string"/>
</simpleType>

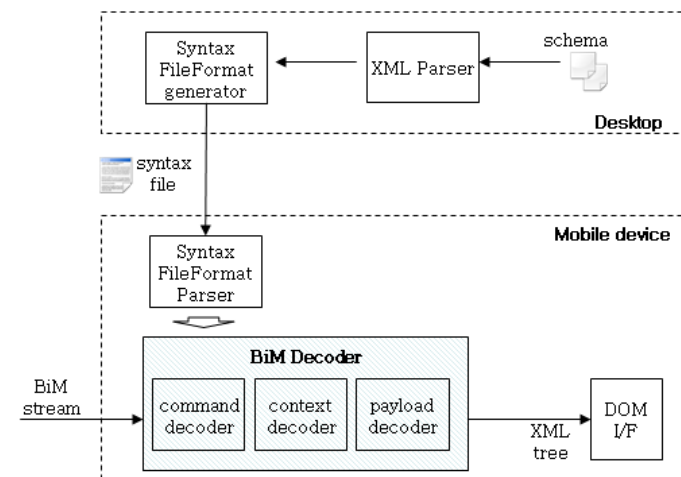
Syntax File :
ST rareType string

Schema :
<simpleType name="addType">
  <union>
    <simpleType>
      <restriction base="string">
        <enumeration value="sum"/>
        <enumeration value="replace"/>
      </restriction>
    </simpleType>
  </union>
</simpleType>
  <simpleType>
    <restriction base="anyURI"/>
  </simpleType>
  </union>

```



<그림 4> 데스크탑 환경에서의 BiM 복호 과정



<그림 5> 모바일 환경에서의 BiM 복호 과정

```

    </simpleType>
  </union>
</simpleType>
Syntax File :
ST addType union(string(enum(sum replace)) anyURI)

```

<그림 7> Simple Type에 대한 Syntax File 생성 과정

을 통하여 정의되기 때문에, 심플 타입에 대한 SF는 이러한 상속관계를 표현하는 트리를 전위 운행하며 텍스트 형태로 기록하는 형식이다. <그림 7>은 이와 같은 Simple Type에 대한 SF 생성의 간단한 예이다.

그 밖에 SF는 타입들 사이의 상속관계를 나타내는 상속 트리 및 네임스페이스 정보 등을 포함하며, 따라서 SF는 스키마 파일과 동등한 정보를 복호기로 전송하게 된다. <그림 8>은 SF의 문법에 대한 BNF 표현이다. SF는 XML에 비하여 가독성을 전혀 고려하지 않고 오직 복호기의 관점에서 스키마를 이해하기 위한 형식이므로, XML에 비하여 용량이 적고 문법이 단순하며 자원을 적게 소비하므로, 특히 XML파서가 동작하기 어려운 소형 모바일 기기 등에서 복호기에 스키마 정보를 입력할 때 효과가 있다. 따라서, 본 연구에서 구현한 복호기는 XML 파싱 모듈에 대하여 종속되지 않으므로써 모바일 플랫폼에서 효과적으로 사용이 가능하다. 이와 같은 본 연구의 복호기는 ESG 메타데이터[6]와 MPEG-4 LAsER[7] 에 대하여 Windows XP, Windows CE, embeded Linux 환경에서 실제로 복호화를 정상적으로 실행하였다.

```

<SF> → <SF_list>
<SF_list> → <namespace>
           | <simpleType>
           | <complexType>
           | <globalElement>
           | <inheritanceTree>
           | <SF_list>
<namespace> → NS <prefix> <uri>
<simpleType> → ST <extendedName>
            <type> <minInclusive>
            <maxInclusive>
<type> → <typeName>
        | <union>
        | <list>
        | <enum>
<union> → union( {<type>}+ )
<list> → list <type>
<enum> → enum( {<type>}+ )

```

```

<complexType> → CT <extendedName>
              {<attributeList>}*
              {<syntaxTreeNode>}*
<attributeList>→ A({<attribute>}+)
<attribute> → (<name> <typeName>
               <isOptional>)
<syntaxTreeNode> → <elementNode>
                  | <groupName>
<elementNode> → E(<extendedName>
                  <typeName>)
               <occurrenceNode>
<occurrenceNode> → O(<minOccur> <maxOccur>)
<groupName> → G({sequence | choice}(
                 <typeName>+)) <occurrenceNode>
<globalElement> → GE <extendedName>
                  <typeName> <minOccur>
                  <maxOccur>
<inheritanceTree> → IN <inheritanceTreeNode>
<inheritanceTreeNode>→
<typeName>({<inheritanceTreeNode>})*

```

<그림 8> SyntaxFile의 BNF

#### 4. 결론

본 연구에서는 모바일 환경의 다양한 플랫폼에서 동작이 가능한 BiM 복호기를 설계 및 구현하였다. 이를 위하여 Syntax File을 정의하여 스키마 파일을 단말에서 직접 파싱하지 않는 것이 가능하도록 하므로써, BiM 복호기만을 사용하여 XML처리가 가능하도록 하였다. 따라서, 본 연구의 복호기는 XML 파서에 대한 종속성을 갖지 않으므로 다양한 모바일 플랫폼에서 효과적인 사용이 가능하다.

#### 참고문헌

- [1] ISO/IEC JTC 1/SC 29/WG 11/N3966 15938-5 "Information Technology Multimedia Content Description Interface PartMultimedia Description Schemes".
- [2] ISO/IEC TR 21000-1 "Information technology - Multimedia framework (MPEG-21) - Part 1: Vision, Technologies and Strategy".
- [3] ISO/IEC FDC 15938-1 FDSI- "Information Technology- Multimedia Content Description Interface - Part 1:Systems", Sydney, July 2001.
- [4] U. Niedermeier, J. Heuer, A. Hutter, W.

Stechele, & A. Kaup, "An MPEG-7 tool for compression and streaming of XML data", *Proc of the 2002 IEEE International Conference on Multimedia and Expo(ICME)*, vol.1 Lausanne, Switzerland, 2002, 521-524.

- [5] World Wid Web XML DOM Working Group, <http://www.w3.org/DOM/>
- [6] *ETSI TS 102 471 V1.1.1* "Digital Video Broadcasting (DVB); IP Datacast over DVB-H: Electronic Service Guide (ESG)".
- [7] *ISO 14496-20* "Information technology - Coding of audio-visual objects - Part 20: Lightweight Application Scene Representation (LAsER) and Simple Aggregation Format (SAF)".