

맞춤형 방송을 위한 TV-Anytime 메타데이터의 이진화

김명훈*, 김혁만*, 양승준*, 김재곤*

*국민대학교 컴퓨터학부, *한국전자통신연구원 디지털방송연구단

*{mhkim,hmkim}@cs.kookmin.ac.kr, *{sjyang,jgkim}@etri.re.kr

Binarization of TV-Anytime Metadata for Personalized TV Services

Myounghoon Kim*, Hyeokman Kim*, Seung-Jun Yang*, Jae-Gon Kim*

*School of Computer Science, Kookmin University, *Broadcasting Media Research Group, ETRI

요 약

본 논문은 디지털 방송 환경에서 TV-AnyTime을 이용한 메타데이터 서비스를 위한 메타데이터 부/복호화 시스템 구현에 관한 논문이다. 부호화 시스템은 생성된 TV-AnyTime 메타데이터의 효율적인 전송을 위해 메타데이터를 TV-AnyTime에 정의된 fragment로 분할하고, 분할된 fragment를 부호화 하는 과정을 포함하고 있다. 또한 fragment의 id와 version을 부여하여 container를 구성하는 것과, 전송된 container의 내용에서 요청된 fragment를 추출하여 처리하는 것을 정의한다. 복호화 시스템은 추출된 fragment들을 분석하고, 사용자의 요청에 따라 해당 fragment를 복호화 한다. 그리고 fragment에 포함된 정보를 이용해 fragment를 관리하는 방법에 대해 정의한다.

1. 서론

디지털 방송 기술은 기존에 접하지 못한 새로운 서비스를 시청자에게 제공 할 뿐만 아니라, 다양한 관점에서 방송 시청 할 수 있는 환경을 제공한다. 그 중에서 메타데이터 서비스는 가장 주목 받고 있는 기술 중의 하나이다.[1] 메타데이터 서비스는 방송 편성표 즉 EPG(Electronic Program Guide), 책의 차례와 같은 형태로 방송 콘텐츠의 내용을 요약하는 TOC(Table of Contents), 방송 콘텐츠의 주요 장면들 나타내는 하이라이트(highlight) 정보 등을 제공 할 수 있다.

메타데이터 서비스를 위해서 방송 콘텐츠에 대한 메타데이터를 방송 표준인 TV-AnyTime를 이용해 XML로 기술 한다[2,3]. 그러나 일반적으로 정형화된 XML은 실제 정보 보다 정보를 표현 하기 위한 텍스트 태그(tag) 때문에 데이터 크기가 매우 커지게 되는 단점이 있다. 따라서 메타데이터를 시청자의 STB(Set-To Box)에 전송 하기 전

에 데이터를 압축하여 부호화하는 과정이 필요하다. 또한 효율적인 전송을 위해서 메타데이터 전체를 한 번에 보내기보다 부분적으로 보내는 방법이 필요하다. 특히 실시간으로 메타데이터를 생성 하는 경우에는 새로 생성된 부분만 전송하거나, 방송 편성표에서 방송 일정의 변화에 따라 부분적 수정이 필요한 경우에는 이미 전송된 데이터중 일부분만 수정할 필요도 있다. 이와 같은 경우를 위해 TV-AnyTime 메타데이터는 전송할 데이터를 fragment 단위로 분할하고, 압축하여 부호화한 후 전송한다. 시청자의 STB는 수신된 메타데이터의 fragment는 복호화 후 기존의 XML에 수정/삭제/추가하게 된다. 이와 같은 과정을 통해 사용자는 방송된 메타데이터를 사용한 메타데이터 서비스를 받을 수 있다.

본 논문은 TV-Anytime 메타데이터의 분할, 부호화, 복호화 모듈 구현에 관한 것이다. 본 논문에서는 메타데이터를 fragment 단위로 분할, 부호화 방법에 대해 정의하고, 부호화된 데이터를 전송 하기 위한 단위와 전송된 데이터를 STB에서 처리 방법을 기술한다.

2. TV-Anytime 메타데이터 서비스 환경

그림 1은 TV-Anytime 메타데이터 서비스 환경의 전체 구조를 나타낸 것이다. 그림 1의 저작 도구에서 생성된 TVA 문서는 TVA 부호화기에 의해 fragment 단위로 분할 후 부호화되고 전송 형태인 container 형태로 재구성된다. container는 방송 송출 장비에서 MPEG-2 transport stream(TS)에 삽입되어 시청자의 STB에 전송된다. 시청자의 STB는 MPEG-2 TS에서 container를 분리하고, 분리된 container는 TVA decoder 의해 fragment로 복호화 된다.

MPEG-2 TS 전송 시스템에서는 특정 container를 식별 하기 위해서 container id와 version을 사용한다. 이 값은 시청자의 STB에서 container를 식별하기 위한 것이고, MPEG-2 TS 전송 방식에 따라 달라진다. 따라서 TV-Anytime에서는 id, version에 대해서는 정의 하고 있지 않다.

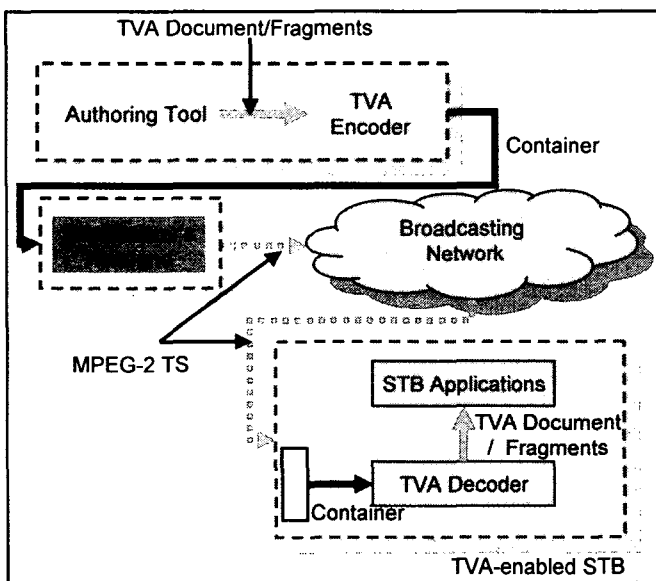


그림 1. TV-AnyTime 시스템

본 논문은 TVA 표준의 메타데이터 스펙을 구현한 것이다. 본 논문에서는 메타데이터 스펙에 정의한 기능중 인덱싱을 제외한 모든 부분을 구현하였다. 스펙에서 정의하지 않았으나 스펙에서 사용하고 있는 MPEG-7 BiM encoder와 decoder 부분은 상용 소프트웨어 모듈인 Expway사의 BinXML 2.1을 사용하였다.[4]

3. TV-Anytime 메타데이터 부호화

TV-Anytime 메타데이터의 부호화기는 TV-Anytime의 스키마에 유효(valid)한 문서와 TV-Anytime 표준에서 fragment 단위로 정의된 TVA fragment을 처리 할 수 있다. TVA fragment는 TVA 문서의 일부분으로서, MPEG-7에서 정의된 FUU(Fragment Update Unit)와 동일한 형태이다.[5] 부호화는 이진화된 fragment를 복호화 과정에서 fragment 단위로 정보를 다루기 위한 fragment id, fragment version을 포함한 container를 생성한다.

그림 2에서 부호화기는 fragment stream generator, BiM encoder, encapsulator로 구성된다. fragment stream generator는 세가지 기능을 포함하고 있다. 첫 번째 TVA-init 메시지를 생성한다. TVA-init 메시지는 복호화에 필요한 Zip library에서 사용할 버퍼 크기, 인덱싱 여부, 스키마 정보를 포함하고, 이 메시지를 통해 BiM encoder 설정 값과 동일한 값으로 TVA 복호화기의 BiM decoder를 설정할 수 있다. 그리고 TVA-init 메시지가 TVA 복호화기로 전송 될 때 TVAMain fragment가 메시지 안에 포함될 수 있다. 만일 메시지에 이 fragment가 포함되어 있지 않다면, 첫 번째 전송할 AU에 이 fragment를 포함해 전송해야 한다. 이 fragment는 TVA 복호화기의 current description tree(CDT)를 초기화 해주는 역할을 한다. 따라서 fragment stream generator는 TVAMain fragment를 포함한 container를 생성할 수 있다. 두 번째 기능은 변화된 fragment를 탐지하여 FragmentUpdateCommand (replace, delete)를 설정하고, ContextPath를 찾고, fragment와 함께 FUU 형태로 BiM encoder에 전달한다. TeM의 형태는 TeM 형태의 AU(Access Unit)를 encapsulator에 전달 한다. 세 번째는 fragment id, version을 부여하고 관리한다.

BiM encoder는 텍스트 형태의 FUU를 압축하여 비트 스트링으로 변화하는 기능을 한다. 이때 압축된 비트 스트링에는 각 fragment 마다 해당 FragmentUpdateCommand와 ContextPath가 부가 되어 BiM 형태의 FUU들을 포함하는 AU를 생성한다.

Encapsulator는 AU와 그 AU에 포함된 FUU들의 fragment id와 version을 입력 받아 이들을 묶어 하나의 container를 생성한다. 이렇게 생성된 container가 TVA

부호화기의 최종 출력이 된다. fragment의 id와 version이 부호화된 FUU와 분리되어 묶어서 생성된다. 이유는, BiM 형태의 경우 TVA 복호화기에서 압축된 FUU를 풀지 않고도 AU에 포함된 FUU들의 fragment들을 식별 하기 위해서 이다.

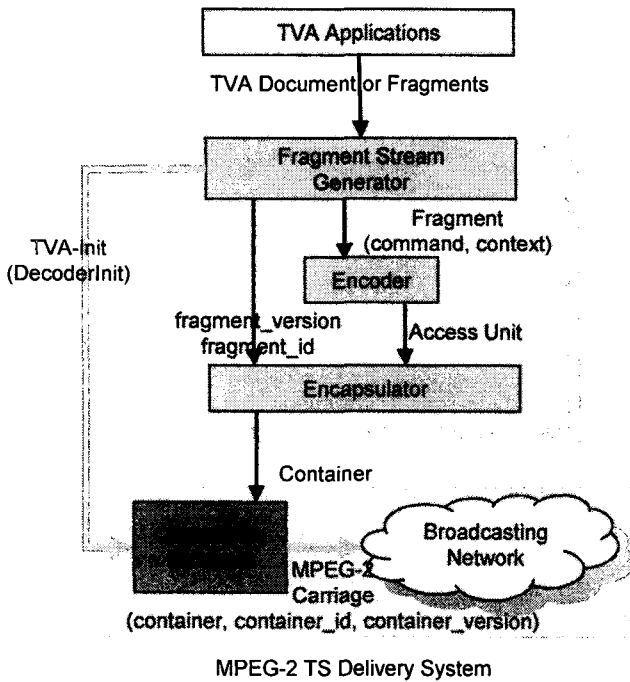


그림 2. 부호화 구조

부호화기는 TVA 문서나 fragment가 입력으로 들어왔을 때 기존과 변화된 fragment를 찾아 TeM 형식의 FUU로 구성된 AU로 생성한다. BiM AU는 TeM AU를 Encoder를 통해 생성하고, BiM AU를 생성한 후 encapsulator를 거쳐 container를 생성한다.

그림 3은 TVA 부호화 주요 모듈 구조를 나타낸 것이다. 이 모듈은 입력이 TVA 문서인 경우 TVA 표준에서 정의된 모든 fragment를 찾아내고, 그 fragment의 루트 엘리먼트 위치 정보(Context Path)를 저장한다. 그리고 Context Path를 이용해 현재 CDT(Current Description Tree)와 비교하여 CDT의 Context Path 위치에 어떤 FragmentUpdateCommand를 적용해야 하는지 결정한다. replaceContent의 경우는 CDT의 fragment의 ID와 현재 fragment의 ID를 이용해 변경 여부를 결정 할 수 있다. 그리고 AU로 구성된 Container는 전송의 효율을 높이기 위해 Container에 포함 될 수 있는 최소 FUU의 개수와 최대 FUU 개수를 제안 할 수 있는 기능을 포함하고 있다.

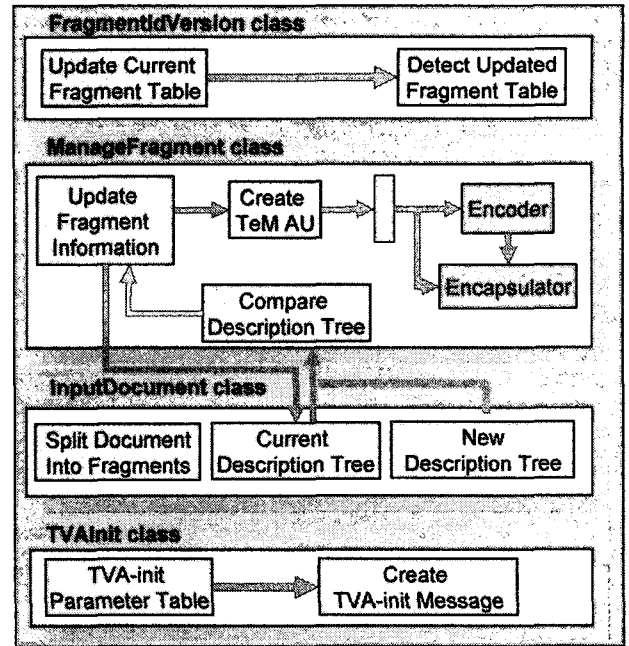


그림 3. 부호화기 주요 모듈 구조

4. TV-Anytime 메타데이터 복호화

그림 4의 TVA 메타데이터 복호화기는 decapsulator, BiM decoder, fragment manager로 구성된다. 복호화기는 TVA-init 메시지 내의 파라미터 초기값을 이용해서 초기화 하고 복호화기의 decapsulator는 전송 받은 container의 AU 부분과 헤더 부분을 분리하는 역할을 한다. 헤더는 AU에 포함된 fragment의 id, version 정보를 포함하고, 이 정보는 fragment manager의 입력이 된다. 그리고 AU는 BiM decoder의 입력이 된다.

BiM decoder는 AU를 입력 받아 TVA-init 메시지로 설정된 복호화 환경의 XML Schema를 이용하여 AU에 포함된 BiM 형태의 FUU들의 복호화 과정을 통해 텍스트 형태의 fragment를 출력하는 기능을 수행한다.

fragment manager는 응용 프로그램의 요청에 따라 응용 프로그램이 원하는 TVA 문서 또는 fragment를 넘겨준다. 이를 위해 먼저 TVA-init 메시지를 받아서 TVA 복호화기의 환경을 설정해야 한다. 이후에 복호화기에 입력되는 container들에서 fragment들과 fragment id, version 정보를 추출하여 메모리에서 관리하여, 응용 프로그램의 요청에 따라 전체 TVA 문서를 전달하거나, 특정 TVA fragment를 원하면 메모리상의 CDT에서 해당 fragment를 검색하여 넘겨준다.

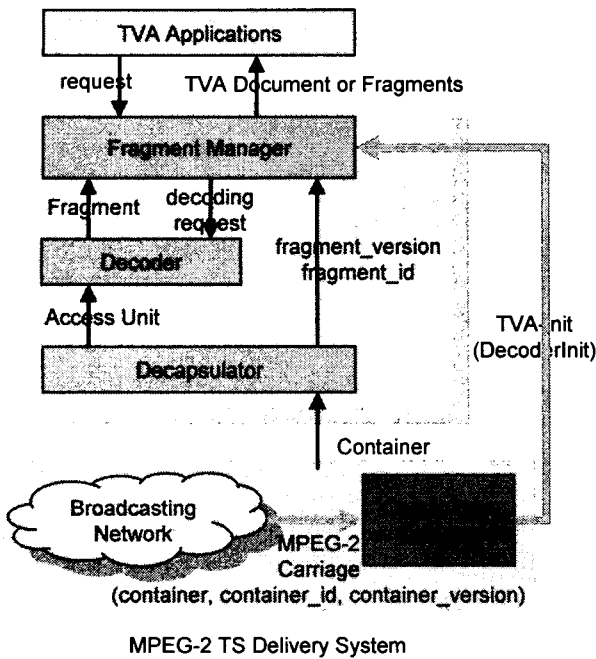


그림 4. 복호화 구조

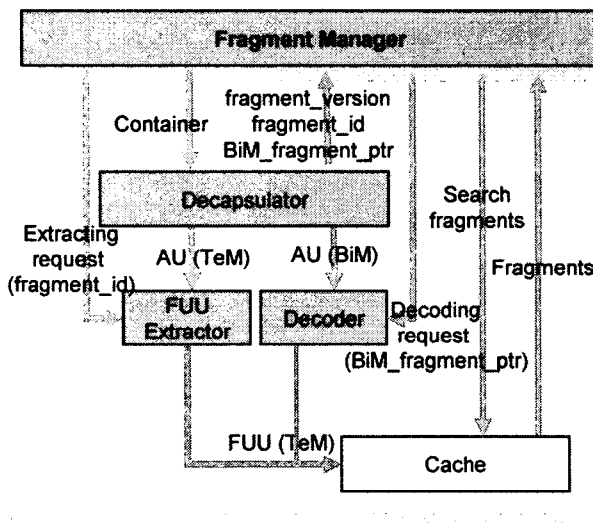


그림 5. 복호화기 주요 모듈

그림 5는 복호화기의 주요 모듈의 기능을 나타낸다. 그림 5에서 fragment manager는 TVA 복호화기 전체를 제어하고 관리하며, 입력되는 container의 FUU를 읽어 들여 CDT를 항상 최신의 정보로 유지하는 역할을 한다. 그리고 decapsulator에서 추출된 fragment id, version 정보를 테이블로 유지하여 선택적으로 BiM decoder에서 복원을 하도록 하는 역할을 한다. 이러한 기능은 복원하는 fragment의 수를 최소화하여 TVA 복호화기의 효율을 높일 수 있다. 복호화기에 container가 입력 될 경우 fragment id, version 정보를 이용하여 decapsulator에서 분리된 AU에서 FUU의 복호화 여부를 결정하고, 복호화

된 fragment를 CDT에 어떠한 명령(add/replace/delete)을 수행 할 지 결정 한다. 그리고 TVA 표준에 fragment는 id는 16비트로 제한되어 있기 때문에 재사용을 위해서 fragment id 테이블에 카운트 값을 할당하고, 일정 기간이 지나면 fragment manager는 fragment를 삭제 하고 id를 재사용한다.

5. 결론 및 향후 과제

본 논문에서는 맞춤형 방송을 위한 TV-Anytime 메타데이터 부호화와 복호화 스펙의 효율적인 구현 방법을 제시하였다. 부호화는 하나의 fragment나, 복수의 fragment를 포함 하는 XML 문서를 부호화 하는 과정을 포함하고, TVA 표준에서 정의 되지 않은 TeM, BiM을 모두 지원한다. 그리고 전송 효율을 높이기 위하여 container에 포함된 fragment의 수를 제어 할 수 있는 기능이 포함되어 있다. 복호화에서는 기존의 존재하는 fragment의 관리 방법과 새로운 fragment가 추가 되었을 때 처리 방법을 제안하였다. 앞으로 TVA 복호화기는 STB에서 동작 해야 하기 때문에 최적화가 필요하고, fragment의 효율적인 검색을 위해 본 논문에서 다루지 않은 인덱싱 스펙의 구현이 필요할 것이다.

[참고 문헌]

- [1] T.J. Sargeant, "Braodcast services enabled by local storage and connectivity", Proc. EUROCON 2003, Vol.2, pp.375-378, Sept. 2003.
- [2] TV-Anytime Forum, "TV-Anytime Phase 1, Part 3: Metadata, Sub-part 1: Metadata schemas", ETSI Standard, ETSI TS 102 822-3-1, V1.1.1, Oct. 2003.
- [3] TV-Anytime Forum, "TV-Anytime Phase 1, Part 3: Metadata, Sub-part 2: System aspects in a uni-directional environment", ETSI Standard, ETSI TS 102 822-3-2, V1.1.1, Oct. 2003.
- [4] Expway, "BinXML-TV™ 2.1", <http://www.expway.com>.
- [5] ISO/IEC JTC1/SC29/WG11 (MPEG), "Information Technology - Multimedia Content Description Interface - Part 1: Systems", International Standard 15938-1, ISO/IEC FDIS 15938-1:2001, Sep. 2001.