

PVR 시스템에서 효율적인 검색을 위한 XML 메타데이터 엔진설계

신은영⁰ 박성한
한양대학교 컴퓨터 공학과
{eyshin⁰, shpark}@cse.hanyang.ac.kr

A XML-based Metadata Engine Design for Effective Retrieval in PVR System

Eunyoung Shin⁰ Sunghan Park
Dept. of Computer Science and Engineering, Hanyang University

요 약

디지털 방송과 함께 저장매체를 갖는 PVR과 셋탑박스가 출현하였지만, 방대한 콘텐츠에 대한 선택의 어려움이 발생하였다. 이러한 문제를 해결하기 위해서 PVR에서는 TV-Anytime과 MPEG-7 표준을 기반으로 멀티미디어 데이터에 대한 메타데이터를 제공한다. 이 메타데이터는 멀티미디어 데이터를 표현하는 특징적인 정보를 포함하고 있어, 콘텐츠에 대한 선택과 검색을 돕는다. 그러나 메타데이터는 그 내용이 방대한 XML document로 구성되어 있어, 효율적이고 빠른 검색이 쉽지 않다. 본 논문은 이러한 XML 메타데이터의 특성을 기반으로 효율적인 검색을 위한 XML 메타데이터 엔진을 설계한다. 제안하는 XML 메타데이터 엔진은 메타데이터의 정보적 특성을 기반으로 인덱싱 구조를 설계하여 XML 메타데이터의 접근 시간을 최소화 한다.

1. 서 론

향후 모든 시청자가 아날로그 방송 환경으로부터 디지털 방송 서비스 환경으로 전환되고 있다. 디지털 방송과 함께 방송 채널은 매우 다양해지면서 방송 시청자는 콘텐츠의 필요요소를 누리게 되었다. 이 때 시청자는 원하는 콘텐츠를 동시에 수용하기를 원하게 되고 이에 디지털 콘텐츠 저장할 수 있는 PVR(Personal Video Recorder)이 등장하게 되었다. 그러나 여전히 시청자로 하여금 프로그램 선별하는데 어려운 문제점이 등장하였다. 이러한 문제를 해결하기 위하여 PVR은 채널 별 프로그램 스케줄 정보에 관련된 메타데이터를 수신하여 처리하는 EPG(Electronic Program guide) 기능을 제공한다. 이러한 EPG 서비스를 가능하게 하기 위해서는 방송 콘텐츠와 콘텐츠에 대한 메타데이터 제공해야 한다. 메타데이터란 멀티미디어 데이터를 기술(description)하는 부가데이터이다. 또한 콘텐츠의 특징적인 내용을 기술하여 내용기반 검색을 가능하게 하는 MPEG-7 또는 TV-Anytime Forum에서 표준화 중인 메타데이터를 포함한다. 메타데이터는 기본적으로 XML document로 이루어지며, 'fragment' 단위로 나뉘어 전송된다.

본 논문은 PVR에서 디지털 방송 영상 데이터를 사용자 요구에 맞게 빠르게 검색하기 위한 XML 메타데이터 엔진을 설계한다. 이 메타데이터 엔진은 XML 메타데이터의 fragment 특성을 기반으로 인덱싱 구조를 구성하게 되는데 메타데이터가 포함하고 있는 정보의 특성에 따라 Node Indexing과 Group Indexing로 구조로 설계한다. 본 논문의 구성은 다음과 같다. 2장에서는 TV-Anytime 표준과 MPEG-7 Description의 Fragmentation 대해 설명한다. 3장에서는 2장에서 설명한 XML 메타데이터의 특성을 바탕

으로 메타데이터 엔진을 설계하고 마지막으로 결론을 내리도록 한다.

2. XML 메타데이터의 특성

2.1. TV-Anytime 표준

TV-Anytime Phase 1에서는 단 방향 방송에 대한 표준이 이미 이루어졌으며 Phase 2에서는 양방향 방송에 대한 표준이 논의 중이다. TV-Anytime의 메타데이터에 대한 표준 내용은 그림 1과 같다.

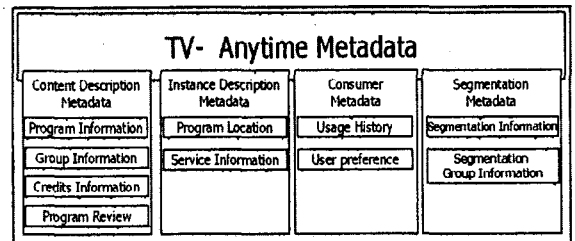


그림 1 TV-Anytime 메타데이터 구조

TV-Anytime의 메타데이터는 Content Description Metadata, Instance Description Metadata, Consumer Metadata, Segmentation Metadata로 분류된다. Content Description은 방송 내용물을 기술하는 Program Information, Group Information, Credits Information, Program Review로 구성된다.

Program Information은 프로그램에 대한 상세 정보를 표현하는

데 제목, 시놉시스, 장르, 언어, 배역 등이 이에 해당한다. Group Information은 프로그램을 그룹화 시킨 정보를 표현하기 위한 series, show, programConcept등이 있다. Credits Information은 프로그램에 대한 감독 배우,작가 등의 출연진 및 제작진의 정보를 표현한다. Program Review Information은 프로그램에 대한 비평 정보를 표현한다.

Instance Description은 Program Location Information, Service Information으로 구성되어 프로그램에 대하여 방송되는 특정 인스턴스 정보를 나타낸다. Program Location Information은 프로그램이 방송되는 시간과 매체에 대한 정보를 표현한다. Service Information은 방송사 이름 URL, 로고, 주로 서비스하는 장르, 아이디 등 방송사 정보를 표현한다.

Consumer metadata는 MPEG-7의 MDS를 기초로 작성되었으며 방송 내용물 측면보다는 이를 소비하는 사용자에 대한 메타데이터를 표현하고, Usage History 및 User Preference로 구성된다. Usage history는 사용자가 방송 내용물에 대한 소비 형태를 나타내는 action history를 표현한다.

Segmentation Metadata는 방송 프로그램의 부분 연속 구간은 지칭하는 "segment"에 대한 메타데이터를 표현한다. Segmentation Metadata는 특정 목적을 위하여 방송 내용물을 시간적인 구간으로 구분해 놓거나, 그룹화 시켜, highlights, scene selections, bookmarks등의 개념을 메타데이터로 표현한 것으로써, 방송 내용물을 사용자가 원하는 방식대로, 비선형적으로 시청할 수 있는 기능을 제공한다. Segmentation information으로 구성된다. Segment information은 세그먼트가 소속한 프로그램의 식별자, 세그먼트 제목, 세그먼트 소속한 프로그램 내에서의 시간 지점 및 지속시간, 키 프레임정보, 식별자 등을 포함한다. Segment group information은 segment들을 하나의 개념으로 그룹핑한 세그먼트 세그먼트들을 표현한다.

2.2 MPEG-7 Description의 Fragmentation

MPEG-7 과 TV-Anytime의 메타데이터는 XML(eXtensible Markup Language)로 기술된다. 그림 2 에서 보는 바와 같이

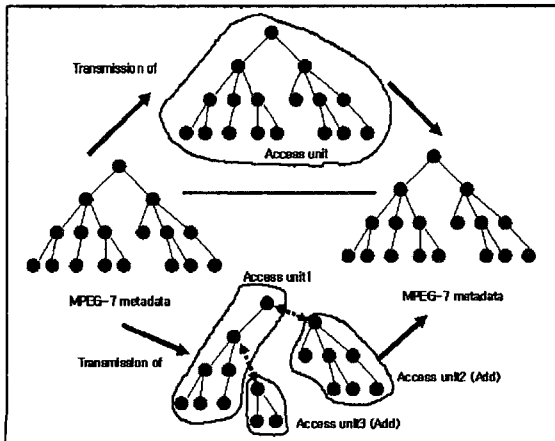


그림 2 MPEG-7 Description과 AU(Access Unit)

메타데이터는 Access Unit(AU)의 형식으로 전송된다. 이때 Access Unit에는 간단한 헤더 정보(FUCommand, FUContext)가 덧붙여진다. 또한 여러 개의 "프래그먼트(Fragment)"라는 단위로 나뉘어지며 계층적인 정보 및 각각의 요소와 속성의 내용을 이진으로 압축 부호화한 형식인 BiM (Binary format for MPEG-7 description streams)으로 전송된다. BiM의 형식은 빠른 전송을 도와 주며, 작은 Fragment단위로 쪼개진 Metadata는 receiver에서 빠르게 검색할 수 있도록 도와준다

3. XML 메타데이터 엔진 설계

메타데이터는 독립적으로 전송, 갱신 접근 할 수 있는 "프래그먼트(Fragment)"라는 단위로 나뉘어 전송된다. 이 전송 받은 프래그먼트를 relational table에 저장한다. 이 프래그먼트의 속성은 fragment_id, fragment_type, fragment_version, fragment_content로 구성된다. fragment_id는 각 프래그먼트의 나타내는 식별자이다. fragment_type은 프래그먼트가 속하는 프래그먼트의 type을 나타낸다. fragment_version은 프래그먼트의 현재 버전정보를 나타낸다. Fragment_content는 text나 binary format으로 된 프래그먼트의 내용이다.

그리고 PVR의 검색 기능에 따라 빠른 검색 결과를 보이기 위해 - 배우, 제목 등 속성으로 프로그램 검색/주요 장면 시청, 장르별 검색 - 두 가지로 index 구조를 구성한다. 첫번째 인덱스는 프로그램 속성으로 원하는 프로그램을 검색하기 위한 Node Index이다. 이 Node index는 key_value와, fragment_id로 구성된다. key_value는 index_key의 value이다. 즉 프래그먼트에 속해 있는 node의 value들이 된다. fragment_id는 프래그먼트의 id로써 fragment table의 fragment_id를 참조한다. 이렇게 Fragment의 각 node의 value들로 구성된 인덱스들은 fragment type 별로 그룹핑 되어 관리되고 access된다. 그리고 각각의 노드 값에 대하여 접근하기 때문에 다중키 indexing 방식을 이용한다. 두 번째 Content Description Metadata의 Group Information과 Segmentation Metadata의 Segment Group Information에 대해서는 Group index를 구성한다. group 정보는 TV 프로그램 장르별 검색이나, 주요 장면검색과 같은 검색에 사용된다. 이 때 사용자는 시청하기 원하는 장르 정보등을 입력하게 되고 PVR에는 장르나 프로그램 세그먼트 그룹에 속하는 리스트 값을 리턴하게 된다. 즉 group검색에서 중요한 정보는 group간의 포함관계를 정의하는 description 정보와 그 그룹에 속한 프래그먼트 리스트 값들이 된다. 그런데 group 정보에 대해 Node Indexing 할 경우 fragment안에 속해 있는 모든 노드에 대해서 인덱싱하게 되므로 메타데이터 엔진에 검색에는 필요하지 않은 정보에 대한 인덱스가 존재하게 된다. 이는 검색의 효율성을 저하시키는 요인이 된다. 또한 그림 3에서의 보는 것과 같이 Group Information Schema에는 Group에 속한 리스트값이 있는 것 아니라 실제 그 프로그램이 속한 그룹에 대한 정보, 즉 MemberOf 엘리먼트로 구성된다. 만약 group 정보에 대하여 Node Indexing을 하고 이 정보에 대하여 검색을 할 경우 Group Information에는 그룹에 속하는 리스트 정보가 없기 때문에 인덱싱된 MemberOf 인덱스와 group fragment에 속하는 모든 노드 값에 대해 인덱스들을 traversing해야 한다. 그러므로 Group information에 대해서

Node indexing을 하지 않고 Group간의 포함관계를 정의 하는 Description 정보, 그 그룹에 속한 Fragment 리스트 값만 추출하여 새롭게 Group Index를 구성한다

```

<complexType name = "GroupInformationType">
  <sequence>
    <element name="GroupType" type="tva:BaseProgramGroupType"/>
    <element name="BasicDescription" type="tva:BaseContentDescription"/>
    <element name="MemberOf" type="tva:BaseMemberOfType" minOccurs="0" maxOccurs="unbounded"/>
  </sequence>
  <attribute name="groupId" type="tva:CRIDType" use="required"/>
  <attribute name="ordered" type="boolean" default="false" use="optional"/>
  <attribute name="numOfItems" type="unsignedInt" use="optional"/>

```

그림 3 TV-Anytime 의 Group Information Schema

Group Index는 *key_name*, *group_id*로 구성된다. *key_name*은 group를 정의하는 이름(description)이며, *group_id*는 그룹을 구별할 수 있는 식별자이다. 또한 그 그룹에 속한 리스트를 저장하는 Group Dictionary를 구성한다. Group Dictionary는 *index_group_id*, *group_member_list*, *num_member*, *group_type*으로 구성된다. *group_id*는 group information fragment의 식별자를 나타낸다. 그리고 *group_member_list*는 이 group에 속한 fragment들의 리스트들을 나타낸다. 여기서 리스트는 Group Indexing안에서 그 그룹에 속하는 리프노드 값들이 된다. *num_member*은 list안의 member 개수를 나타낸다. *group_type*은 index중 group information의 fragment의 type 정보이다

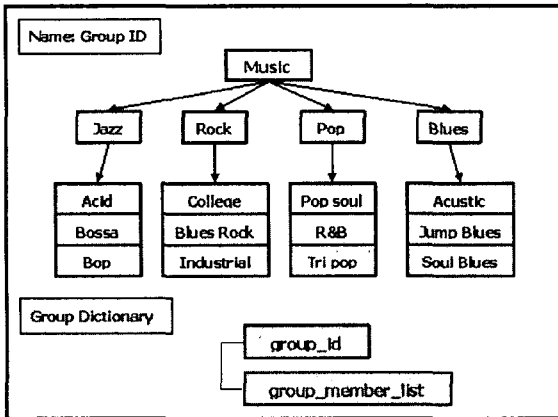


그림 4 Group index 구조도 예

결과적으로 Group Index는 그림 4와 같이 group을 정의 하는 Group Description에 대해서 계층적으로 indexing 하고 그 Group에 속하는 List를 저장하는 테이블로 구성하게 된다

메타 데이터 엔진에는 여러 개의 index가 존재한다. 그러므로 어떤 인덱스를 선택할 것인가를 결정하기 위해 두 개의 보조 테이블을 만든다 보조테이블은 Fragmenttype Dictionary와 Node Dictionary 두 개로 구성된다. 첫째로 Fragmenttype Dictionary는 *index_id*, *fragment_type*, *num_key_node*, *key_node_list*, *index_version*으로 구성된다. 여기서 *index_id*는 index의 식별자이고, *fragment_type*은 index를 만드는 fragment의 type 정보이며, *num_key_node*는 index key를

구성하는 key node의 개수이다. 그리고 *key_node_list*는 프래그먼트 안에 속해 있는 key node의 list이다. 또한 *index_version*은 fragment의 version 정보를 나타낸다. 즉 이 version 정보는 sender에서 보내는 fragment의 version 정보와 비교하여 update를 할 것인지를 결정하는데 중요한 정보를 제공한다. 둘째로 Node Dictionary는 Index들에서 쓰이는 *key_node*들의 정보이다. 각각은 *node_id*, *node_path*, *node_type*으로 구성된다. 이때 *node_id*는 각 node의 식별자이고 *node_path*는 절대 경로로 각 노드들의 path를 나타내며, *node_type*은 각 노드들의 type을 나타낸다. 그림 5은 제안한 메타데이터 엔진의 전체 구성도를 보여준다.

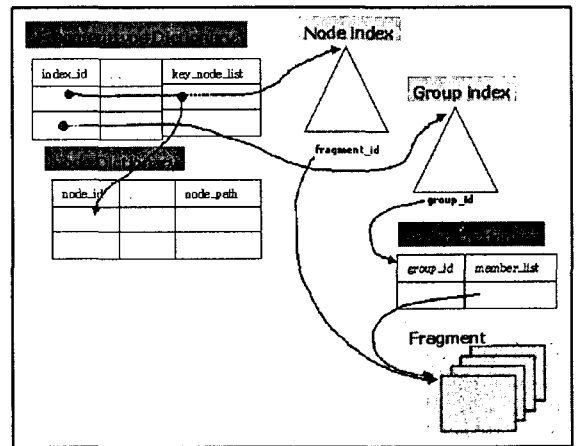


그림 5 Metadata Engine의 전체 구성도

4. 결론

본 논문에서는 PVR system에서 프래그먼트를 기반으로 한 XML 메타데이터 엔진 설계를 제안하였다. 이 엔진은 메타데이터에 효율적으로 접근하기 위한 인덱스 구조 설계하였다. 또한 이 엔진의 디자인은 관계 데이터베이스 시스템으로 구축할 수 있고 인덱스는 B-tree들로 구현할 수 있기 때문에 쉽게 구현 가능하다는 장점을 갖는다.

참고문헌

- [1] TV-Anytime Forum, "Specification Series: S -3 on Metadata, Part A" version 1.3, December 15, 2002.
- [2] TV-Anytime Forum, "Specification Series: S - 3 on Metadata, Part B" version 1.3, December 15, 2002.
- [3] ISO/IEC 15983-5, Information Technology - Multimedia content description interface - Part 5: Multimedia content description Schemes, 2001.
- [4] H. Shin, "A Storage and Retrieval Method of XML-based Metadata in PVR Environment", IEEE Transactions on Consumer Electronics, Vol. 49, No 4, NOVEMBER 2003.
- [5] U. Westerman, W. Klas, " An Analysis of XML Database Solutions for the Management of MPEG-7 Media Descriptions" , ACM Computing Surveys, Vol. 35, No4 December 2003 pp.331-373.