

# TV-Anytime기반 메타데이터 관리 시스템

이민우<sup>○</sup> 박종현 이용희 정민옥 강지훈  
충남대학교 컴퓨터학과  
{ cslmw<sup>○</sup>, jhpark, lyhcool, ultra999, jhkang }@cs.cnu.ac.kr

## The metadata management System based on TV-Anytime

Min-Woo Lee<sup>○</sup>, Jong-Hyun Park, Yong-Hee Lee, Min-Ok Jung, Ji-Hoon Kang  
Dept. of Computer Science, Chungnam National University

### 요 약

TV-Anytime은 차세대 디지털 방송을 위한 표준으로 기존의 방송 시스템에서 방송 서비스 제공자가 사용자에게 방송 프로그램을 일방적으로 전송하는 것과는 달리 사용자가 다 채널 환경하에서 관련 메타데이터를 이용하여 채널을 직접 검색하고, 검색한 정보를 저장하였다가 시간과 장소에 무관하게 사용하는 것을 목표로 한다. 이러한 환경을 가능하게 하기 위하여 본 논문에서는 TV-Anytime 메타데이터를 저장 및 관리하고 검색하기 위한 시스템을 설계 및 구현한다.

TV-Anytime기반 메타데이터 관리 시스템은 메타데이터의 특성을 파악하여 일반적인 메타데이터 관리 방법 보다는 TV-Anytime 메타데이터에 특화된 방법으로 설계하고 구현하였다. 또한, 사용자에게 보다 풍부한 질의를 제공하기 위해서 MPEG7을 추가하여 내용기반 검색을 가능하도록 할 뿐만 아니라, XQuery를 검색을 위한 질의어로 사용하여 시스템간의 상호운용성을 보장하였다.

### 1. 서론

디지털 방송은 차세대 방송의 새로운 패러다임으로써, 디지털 방송의 시작으로 인해 채널의 수가 증가하고 콘텐츠의 양이 방대해 지게 되었다. 따라서 사용자가 원하는 채널이나 콘텐츠를 빠르고 효율적으로 검색하고 이용할 수 있도록 하는 서비스가 필요하게 되었다. 이러한 서비스를 가능하게 하기 위하여 현재 TV-Anytime Forum[1]에서는 차세대 디지털방송용 메타데이터 표준인 TV-Anytime 메타데이터를 제안하고 있다.

TV-Anytime[3] 메타데이터는 단일의 XML 스키마를 기반으로 기술하고, 그 양이 대용량이며 멀티미디어 데이터에 대한 MPEG7[2] 메타데이터를 포함하고 있다. 본 논문은 이러한 특성을 고려하여 TV-Anytime 메타데이터를 보다 효율적으로 저장 및 검색하기 위한 시스템을 제안하고 있다. 그리고 검색 항목 및 반환 항목을 직접 지정할 수 있는 XML 데이터의 검색을 위한 국제 표준 질의 언어인 XQuery를 질의 언어로 이용하여 시스템 간의 상호운용성을 높이고록 제안하고 있다. 또한 사용자에게 보다 풍부한 질의를 제공하기 위하여 MPEG7의 Visual 부분을 TV-Anytime과 연계하여 내용기반 검색을 가능하게 한다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구들을 살펴 보며, 3장에서는 본 메타데이터 관리 시스템의 구조를 기술하고, 4장에서는 결론 및 향후 과제에 대하여 논한다.

### 2. 관련 연구

XML 문서를 효율적으로 저장 및 검색하기 위한 여러 연구가 이루어져 왔다. 특히 구조화된 XML문서를 관계형 데이터베이스에 저장하여 필요한 정보의 검색 및 관리의 편의성을 돕고자 하는 연구가 활발히 진행중이다. 또한 필요한 정보의 검색을 위해 XML Query Language로써 XQuery를 채택하고 있다. 따라서 최

근의 연구들은 XQuery를 이해하고 처리할 수 있을 뿐만 아니라 실제 내용에 접근하기 위하여 관계형 데이터베이스에서 처리할 수 있는 SQL질의로 변형하는 시스템을 개발하고 있다.

그중 대표적인 시스템으로써 Agora[4], EXIP[10], XPERANTO[6,7] 등이 있다. 이러한 시스템들은 정보 소스에 대한 뷰를 제공하는 wrapper를 두고 있다. 사용자는 검색 조건을 입력하고 입력된 검색 조건은 이러한 뷰를 기반으로 XQuery로 구성이 된다. XQuery는 파서를 통하여 분석이 되고 분석된 정보를 기반으로 SQL로 변환하여 이러한 SQL문을 가지고 실제 DBMS에 질의하여 원하는 결과를 생성한다. 이러한 시스템들은 Optimizer의 도입으로 XQuery에 대하여 최적화를 수행하고 이를 SQL문으로 변환한다는 점이 본 시스템과 유사하다. 특히 EXIP 시스템은 입력 XQuery를 분석하여 Abstract Syntax Tree를 구성하고 이러한 AST를 기반으로 실제 DBMS에 질의 가능한 언어로 변환한다는 점이 본 시스템의 XQuery Syntax Tree와 유사하다고 하겠다.

또 다른 XML 질의 처리 시스템으로써 XQP[9] 시스템이 있다. XQP 시스템은 Element 태그의 인코딩(ET-encoding)과 Element 내용의 인코딩(EC-encoding)을 사용하는 XML 데이터를 묘사한다. 입력 XQuery문에 표현된 패스 표현은 ET-encoding path라고 하는 인코딩의 연속으로써 표현하고 또한 유사하게 XML 데이터에 패스 표현을 적용함으로써 EC-encoding path를 표현한다. XQP 시스템은 전형적인 XQuery 문을 수락하는데, 그것은 제한사항을 부과하기 위한 selection 구문과, 임의의 Element의 내용을 반환하기 위한 projection 구문으로 구성되어 있다. XQuery문을 처리하기 위해 XQP 시스템은 우선 각각 selection 제한 사항과 projection 요구사항을 만족하는 EC-encoding path를 검색할 것이다. 그리고 정확한 데이터를 획득하기 위해 Correct Ancestor Encoding을 획득한다. 이러한 접근

방법은 본 시스템의 검색 패스와 반환 패스의 공통 패스를 획득하고 이들간의 Dewey Number 비교를 통해 정확한 Element를 선정하는 과정과 유사하다고 할 수 있다. 그러나 XQP 시스템은 단 하나의 selection 구문과 projection 구문만을 고려했다는 단점이 있는데 반해 본 시스템은 여러 개의 selection 구문과 projection 구문을 가진 XQuery를 처리할 수 있다는 장점이 있다.

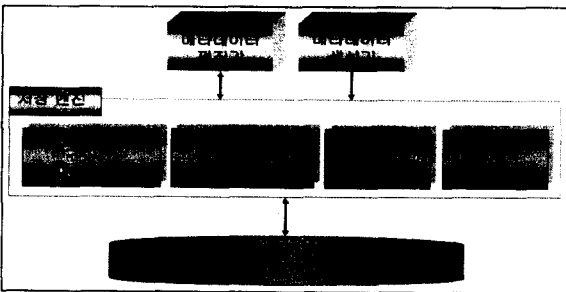
그러나 이들 시스템 모두 일반적인 XML 데이터의 관리목표로 하므로 본 논문에서와 같이 단일의 메타데이터를 관리하기 위해서는 특화된 방법이 필요하다.

### 3. TV-Anytime 메타데이터 관리 시스템

본 메타데이터 관리 시스템은 크게 메타데이터의 저장을 위한 메타데이터 저장 엔진과 검색을 위한 검색 엔진으로 구성되어 있다. 본 시스템의 저장 엔진에서는 메타데이터를 DBMS에 효율적으로 저장하기 위한 DB 스키마 구조를 [그림 3-2]와 같이 정의하였으며 내용 기반 검색을 위하여 MPEG7 Visual Descriptor Scalable Color Descriptor와 Homogeneous Texture Descriptor 만을 별도로 정의하여 사용하였다. 그리고 메타데이터 검색을 위한 질의 언어로 XQuery를 채택하여 검색 프로토콜을 구성하였다. 그러므로 검색 엔진 하부 구조에 상관없이 질의가 가능할 뿐만 아니라 서버가 XQuery를 이해할 수 있다면 클라이언트 모듈이 독립적으로 사용될 수 있다는 장점을 가지고 있다.

#### 3.1 저장 엔진 구조

디지털 방송용 대용량 메타데이터 관리 시스템에서 메타데이터로 사용되는 TV-Anytime 및 MPEG-7 메타데이터의 효율적인 저장과 관리를 위하여 사용되는 메타데이터 저장 엔진의 구조는 [그림 3-1]과 같다.



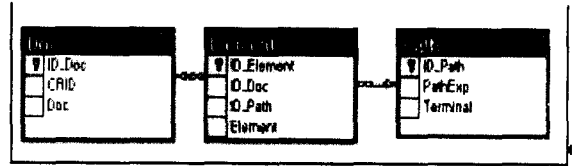
[그림 3-1] 메타데이터 저장 엔진의 구조

본 논문에서는 대용량의 TV-Anytime 및 MPEG-7 메타데이터의 저장과 관리를 위하여 대용량 데이터의 효율적인 저장과 관리가 가능한 상용 데이터베이스를 사용한다. 메타데이터 저장 엔진은 TV-Anytime 및 MPEG-7 메타데이터의 삽입, 삭제 그리고 갱신을 위하여 InsertDoc과 InsertMPEG-7 Doc, DeleteDoc과 DeleteMPEG-7 Doc, UpdateDoc 그리고 GetDoc 이라는 여섯 개의 모듈로 구성된다.

#### 3.2 저장 엔진 DB 스키마

본 논문에서는 TV-Anytime 메타데이터를 위한 스키마를 분석하여 Edge 단위로 동일한 엘리먼트의 이름을 가지는 모든 노드는 동일한 Table에 저장하는 Binary방법을 사용하였고 또한, 각각의 엘리먼트 마다 고유한 노드의 ID를 부여하기 위하여 Dewey 방법[5]을 사용하여 단일 문서에서 모든 엘리먼트 노드는 서로 다른 ID 즉, 고유한 키를 갖도록 하였다. Dewey ID는 크게 두 가지로 사용된다. 첫째, 데이터베이스 스키마에서 부모 자식 테이블간 연결을 맺는 포인터의 역할을 하여 분해된 문서를 쉽게 재조합 할 수 있다. 둘째, Dewey ID는 루트 노드부터 각 노

드까지의 경로이므로 간단한 질의표현으로 정확한 질의결과를 얻을 수 있다.

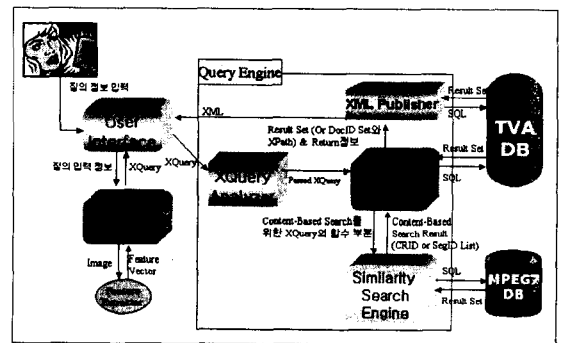


[그림 3-2] Element, Path, Document Table의 관계

[그림 3-2]는 각 Element, Path, Document Table의 관계를 나타내고 있다. TV-Anytime 메타데이터의 기술에 사용되는 모든 엘리먼트 노드들은 동일한 이름을 가지는 노드끼리 동일한 Table에 저장되고 모든 Element Table은 [그림 3-2]와 같이 Path Table과 Document Table과 연결되는 구조로 데이터베이스 스키마는 구성되어 있다. Path Table의 PathExp 필드는 각 엘리먼트 노드의 Path를 저장하여 사용자가 질의한 XQuery의 XPath 표현을 한번의 비교만으로 직접 해당 Table을 선택할 수 있도록 함으로써 검색 시간을 줄일 수 있다.

#### 3.3 검색 엔진 구조

[그림 3-3]은 메타데이터 검색 시스템의 구조이다.



[그림 3-3] 메타데이터 검색 엔진의 구조

사용자가 인터페이스를 통해 찾고자 하는 질의어를 입력할 경우 입력된 질의어는 XQuery Generator를 통해 XQuery 질의어로 구성되며 내용 기반 검색 정보가 존재할 경우 Feature Extractor를 사용하여 이미지에 관한 특성을 추출하고 기존 정보와 함께 XQuery를 구성한다. XQuery는 서버측으로 제공되며 우선 XQuery Analyzer는 XQuery Syntax Tree를 생성하고 이 Tree를 순회하면서 구조 정보를 추출한다. XQuery Engine은 이 구조 정보를 기반으로 SQL문을 자동 생성한다. XQuery 구조 정보에 내용 기반 검색 정보가 존재할 경우 이 정보를 Similarity Search Engine으로 제공하며 질의 이미지 특성 정보와 데이터 베이스에 저장되어 있던 이미지 정보를 비교하여 가장 유사한 이미지의 CRID와 Segment ID를 반환한다. 이 2가지 정보와 키워드 기반 검색 정보를 통합하여 SQL문을 구성하여 TVA DB에 질의하고 SQL Result Set을 받아들인다. SQL Result Set과 Return구조 정보가 XML Publisher로 제공되며 이 정보를 기반으로 사용자에게 반환할 XML 문서를 재구성하여 사용자에게 반환한다.

#### 3.4 내용 기반 검색

이미지 데이터에 대한 Color 검색, Texture 검색, Color와 Texture 혼합 검색을 위하여 본 논문에서는 ImageSearchForColor(), ImageSearchForTexture(), ImageSearchForMixed()를 정의하며, 사용자 정의 함수 내에 유사도 검색을 위하여

SimilarTo() XQuery 함수를 정의하여 사용한다. 다음 그림 [3-4]는 프로그램 제목이 "원더러스"를 포함하고 있고 프로그램 CRID와 이미지에 대한 segmentID가 질의 이미지의 Color와 가장 유사한 이미지의 CRID 및 SegmentID가 같은 문서의 ProgramInformation을 반환하려는 XQuery문이다.

```

define function ImageSearchForColor() as element()
{
  <ColorResults>{
    for $d in input("TVMP7")
    for $ImageDI in $d/Mpeg7ForTVATime/DescriptionImage
    for $ImageCRID in $ImageDI/@CRID
    for $ImageSegmentId in $ImageDI/@SegmentId
    for $ImageNumOfCoeff in
      $ImageDI/ScalableColor/@numOfCoeff
    for $ImageNumOfBitPlanesDiscarded in
      $ImageDI/ScalableColor/@numOfBitPlanesDiscarded
    for $ImageCoeff in $ImageDI/ScalableColor/Coeff
    for $Similarity in SimilarTo(string($ImageNumOfCoeff), "64",
      string($ImageNumOfBitPlanesDiscarded), "0",
      string($ImageCoeff), "34 -14 -38 ... -3 -4 -2 -7 -7")
    order by $Similarity
    return
    <ColorResult> { $ImageCRID, $ImageSegmentId } </ColorResult>
  } </ColorResults>
}
<Results>{ for $d in input("TVAnyTime") return <Result>{
distinct-values(
  for $p1 in $d/TVMain/ProgramDescription
  for $p2 in $p1/ProgramInformationTable/ProgramInformation
  for $p3 in $p2/BasicDescription/Title
  for $p4 in
    $p1/ProgramLocationTable/OnDemandProgram/Program/@crid
  for $p5 in
    $p1/SegmentInformationTable/SegmentList/SegmentInformation/@segmentId
  for $CRS in ImageSearchForColor()
  for $SCR in $CRS/ColorResult[position()<=1]
  for $ImageCRID in $SCR/@CRID
  for $ImageSegmentID in $SCR/@SegmentId
  where contains(string($p3), "원더러스") and ($p4=$ImageCRID and
    $p5=$ImageSegmentID)
  return $p2
} } </Result> } </Results>
    
```

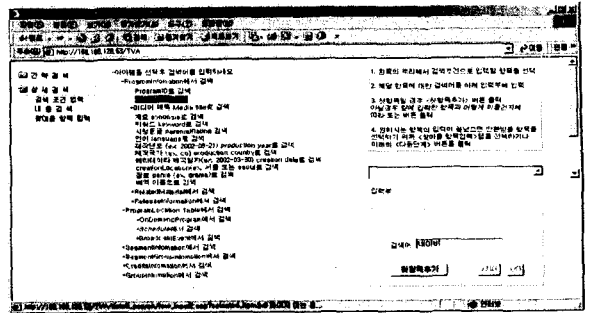
[그림 3-4] 키워드 및 이미지 Color 혼합 검색 XQuery

본 시스템에서는 SR-Tree[8]를 이용하여 이미지에 대한 색인을 구현하였다. SR-Tree는 다른 구조에 비해 적은 공간으로 point를 나누기 때문에 disjointness가 향상되고, nearest neighbour queries performance가 증가 되므로 검색에 알맞은 구조라 할 수 있다. 또한, 대용량의 이미지를 처리하기에는 메모리에 부담을 주기 때문에 트리 인덱스는 파일로 구성하여 사용한다. 그러므로 이미지의 삽입, 업데이트, 삭제가 DB와 index 파일에서 동시에 이루어 질 수 있도록 하였다.

3.5 사용자 인터페이스

XQuery는 XML을 위한 표준 질의 언어로서 XML 데이터의 저장 형식에 무관하게 질의 할 수 있는 장점을 가진 반면, 초보자가 사용하기에 어려운 복잡한 구조를 이루고 있다. 따라서 XQuery에 익숙하지 않은 사용자도 질의 가능하도록 [그림 3-5]와 같은 인터페이스를 제공한다.

본 시스템의 인터페이스는 와 같이 TV-Anytime 메타데이터를 위한 스키마를 분석하여 최대한 유사하게 각 검색 항목을 Tree 형식으로 표현하였다. 따라서 메타데이터의 구조를 간접적으로나마 파악할 수 있으며 또한 검색의 편의성을 제공한다는 장점을 지니고 있다. 또한 질의 이미지 입력을 통해 내용 기반 검색도 가능하도록 하고 있다.



[그림 3-5] 메타데이터 검색을 위한 인터페이스

4. 결론

본 논문에서는 TV-Anytime과 MPEG7 메타데이터를 저장, 검색 및 관리하기 위한 시스템을 제안하고 이를 구현하였다. 이는 방송용 메타데이터의 저장과 검색뿐만 아니라 XML 데이터의 저장과 검색을 위한 기존의 연구와 상호 보완적 성격을 가지므로 서로에게 도움을 줄 수 있을 것으로 사료된다. 또한, 본 논문에서 개발한 XQuery 엔진은 XML데이터를 검색하기 위한 표준 질의어를 사용하였으므로 방송용 메타데이터의 검색뿐만 아니라 XML 데이터의 검색을 위한 연구에 도움이 될 것이다.

향후, 사용자에게 보다 풍부한 질의를 가능하게 하기 위하여 MPEG7의 추가적인 확장과 TV-Anytime 메타데이터 Version의 갱신을 위한 연구가 필요하다.

5. 참고 문헌

- [1] IT Forum Korea 2002, "TV-Anytime Forum 최근 동향" May 2002.
- [2] Overview of the MPEG-7 Standard, Dec. 2001.
- [3] TV-Anytime Specification Series, August, 2003
- [4] I. Manolescu, D. Florescu, D. Kossman, "Answering XML queries over heterogeneous data sources", Proc. VLDB Conf. Roma, Italy, Sept. 2001.
- [5] I. Tatarinov, S.D. Viglas, K. Beyer, J. Shanmugasundaram, E. Shekita, & C. Zhang, "Storing and Querying Ordered XML Using a Relational Database System," Proc. ACM SIGMOD Conf., June 2002.
- [6] J. Shanmugasundaram, "XPERANTO: Bridging Relational Technology and XML", IBM Research Report, June 2001.
- [7] M. Carey, J. Kiernan, J. Shanmugasundaram, E. Shekita, & S. Subramanian, "XPERANTO: Middleware for Publishing Object-Relational Data as XML Documents," Proc. VLDB 2000, pp 646-648, September 2000.
- [8] N. Katayama, S. Satoh, "The SR-tree: An Index Structure for High-Dimensional Nearest Neighbor Queries," Proc. ACM SIGMOD Conf, pp.369-380, Tucson, Arizona, May 1997.
- [9] Y.-H. Chang, C.-T. Lee, "Supporting Selection-Projection XQuery Processing Based on Encoding Paths," Proc. 8th International DASFAA '03 Conf., Kyoto, Japan, March 2003.
- [10] Y. Papakonstantinou, V. Vassalos, "Architecture and Implementation of an XQuery based Information Integration Platform." IEEE Data Engineering Bulletin, Vol. 25, No. 1, March 2002.