

데이터베이스를 이용한 TV-Anytime 메타데이터의 저장

김병규⁰, 박종현*, 강지훈*
한국과학기술정보연구원, 충남대학교 컴퓨터과학과*
yourovin@kisti.re.kr, fjhpark, jhkang}@cs.cnu.ac.kr

Storing TV-Anytime Metadata using Databases

Byung-Kyu Kim⁰, Jong-Hyun Park*, Ji-Hoon Kang*
Korea Institute of Science and Technology Information
Dept. of Computer Science, Chungnam National University*

요 약

TV-Anytime은 차세대 디지털 방송을 위한 메타데이터 표준으로 XML 형식으로 기술된다. TV-Anytime 메타데이터는 그 특성상 단일 스키마를 따르며, 규모에 있어서 대용량이다. 그러므로 데이터베이스를 이용하여 메타데이터를 저장하고 관리할 필요가 있다. 본 논문에서는 대용량 TV-Anytime 메타데이터를, 데이터베이스를 이용하여 저장하고 관리하기 위한 저장 엔진을 제안한다. 단일 스키마에 기반을 두고 있다는 점을 고려하여 TV-Anytime 메타데이터의 구조에 적합한 데이터베이스 스키마를 설계하였다. 문서 재조합을 효율적으로 할 수 있도록 노드 번호를 위하여 Dewey 방법을 사용하였으며, 조인 연산을 줄이기 위하여 Path 테이블을 두었다. 저장 엔진은 TV-Anytime 관리 시스템에서 XQuery 엔진과 같은 XML 질의처리기와 연동될 수 있다.

1. 서론

TV-Anytime[1] Forum은 소비자가 원하는 시간에 어느 곳에서든지 원하는 프로그램을 시청 할 수 있도록 하기 위한 방송용 표준 메타데이터인 TV-Anytime을 제안하고 있다. TV-Anytime 기술은 차세대 디지털 방송용 메타데이터 국제표준으로 자리잡아가고 있는 기술로 점점 대용량화 되고 저해지는 저장장치를 이용하여 사용자가 원하는 멀티미디어 데이터를 저장하였다가 시간과 장소에 구애 받지 않고 사용할 수 있도록 하는 기술이다.

TV-Anytime 메타데이터의 특성을 살펴보면 첫째, 방송용이라는 특성상 그 크기가 대용량이고 두번째는 단일의 스키마를 갖고 문서 형식이 XML이다. 이러한 특징을 갖는 TV-Anytime을 효율적으로 저장·관리하기 위해서는 데이터베이스를 이용하는 것이 매우 효과적이라 할 수 있다.

본 논문에서는 대용량 TV-Anytime 메타데이터를, 데이터베이스를 이용하여 저장하고 관리하기 위한 저장 엔진을 제안한다. 단일 스키마에 기반을 두고 있다는 점을 고려하여 TV-Anytime 메타데이터의 구조에 적합한 데이터베이스 스키마를 설계하였다. 문서 재조합을 효율적으로 할 수 있도록 노드 번호를 위하여 Dewey 방법을 사용하였으며, 조인 연산을 줄이기 위하여 Path 테이블을 두었다. 저장 엔진은 TV-Anytime 관리 시스템에서 XQuery 엔진과 같은 XML 질의처리기와 연동될 수 있다. 본 논문의 구성은 다음과 같다. 제 2절에서는 관련연구로써 본 연구와 비교될 수 있는 기존의 XML 문서 저장기법을 살펴본다. 제 3절에선 저장 엔진을 위한 DB 스키마의 설계내용과 저장엔진

의 구조 및 저장엔진의 구현에 관한 내용을 기술한다. 마지막으로 제 4절에서 결론을 기술한다.

2. 관련 연구

TV-Anytime 메타데이터의 형식은 XML이다. 구조화된 XML문서를 Flat한 구조의 관계형 데이터 베이스에 저장하기 위한 방법은 크게 DTD와 같은 스키마 정보가 있는 경우와 그렇지 않은 경우로 나누어 볼 수 있다.

2.1 Instance기반 저장 기법

스키마 정보가 없는 경우에는 문서로부터 스키마 정보를 추출해 내어 관계형 테이블들을 생성하는 방법과, 스키마 정보는 무시하고 XML문서를 구성하는 요소들, 즉 노드와 간선(Edge)들만을 저장하는 방법이 있다. 전자의 경우는 STORED[2]에 해당하고, 후자는 Binary방식[3]등이 대표적이다.

Binary 방식은 간선의 이름 별로 별도의 이진 테이블을 만드는 방법이며 이렇게 만들어진 모든 테이블들을 outer join으로 결합시키는 것이 Universal방식이다.



* 이 연구는 HK21 충남대학교 정보통신 인력 양성 사업단의 지원에 받았음

[그림 1] 트리 형태의 XML문서와 Binary방식

Binary방식은 XML문서와 함께 주어진 스키마 정보를 전혀 이용하지 못하므로 검색시 많은 테이블들의 조인 연산으로 인해 필연적으로 성능의 문제를 제기한다.

2.2 DTD 기반 저장 기법

스키마 정보를 이용하는 경우는 DTD로부터 관계형 스키마를 생성하는 방법으로 Inlining 방법[4]이 있다. Inlining 방법은 DTD에서 표현될 수 있는 내용들 중에서 실제 관계형 테이블들을 만드는데 사용될 수 있는 정보만을 이용하여 DTD, Element Graph를 만들고, Element Graph를 순회 하면서 관계형 테이블들을 만드는 기법이다.

Inlining 방법은 크게 세가지로 나뉘는데, Basic Inlining은 Element Graph를 깊이 우선으로 탐색하면서 *를 만나거나 사이클을 만나지 않는 한, 단말에 있는 엘리먼트들을 모두 하나의 테이블에 인라인 시키는 것이다. 다만, *나 사이클을 만나게 되면 독립된 테이블로 만들게 된다. 그러나, Basic Inlining은 너무 많은 수의 테이블을 만들어 내는 단점이 있다. 이러한 문제점을 해결하기 위해 제안된 Shared Inlining은 루트 노드에 해당하는 엘리먼트에서만 테이블을 생성하게 된다. 서브 엘리먼트 노드들을 방문하면서 엘리먼트들을 인라인시키는 것은 Basic과 같지만, Shared에서는 추가적으로 다른 테이블과 공유되어질 수 있는 엘리먼트를 만나게 되면 인라인 시키지 않고 독립된 테이블로 처리하게 되는 점이 Basic과 다르다. 끝으로 Hybrid는 Shared에서 인라인되지 않은 엘리먼트 중에서 *로 연결되지 않고 재귀적으로 연결된 경우가 아니면 다른 테이블에 인라인시키는 방법으로 Shared의 경우보다 조인의 횟수를 줄임으로써 성능 향상 효과가 있다.

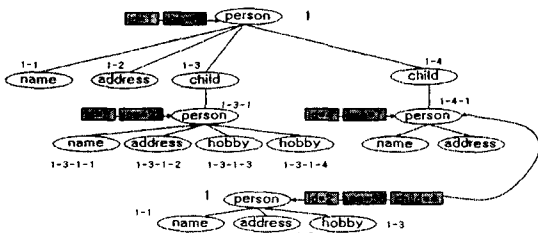
위와 같은 inlining기법은 요구되는 DTD에 맞게 DB 스키마를 정의하는 작업이 쉽지 않으며 검색의 결과로 복잡한 XML 문서의 생성을 요구할 때 다루기 어려운 단점이 있다.

3. TV-Anytime 메타데이터 저장 엔진

본 장에서는 TV-Anytime 메타데이터 저장을 위한 DB 스키마 설계와 저장 엔진의 구조 및 각 요소들을 기술한다.

3.1 저장 엔진 DB 스키마

TV-Anytime 메타데이터는 단일의 XML Schema를 따르고 대용량이라는 특성을 가진다. 이러한 특성을 감안하여 저장 엔진에서는 대용량의 메타데이터를 효율적으로 저장하기 위하여 상용 관계형 데이터 베이스를 사용하였고, 단일의 TV-Anytime Schema를 분석하여 DB Schema를 설계하였다.



[그림 2] Dewey 방법

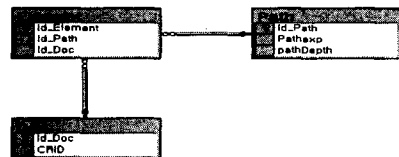
본 논문에서는 TV-Anytime 메타데이터 Schema를 이용하여 동일한 Element를 미리 정의된 엘리먼트 Table에 저장하는 방법을 사용하였고 각각의 Element마다 고유한 Node의 ID를 부여하기 위하여 [그림 2]와 같이 Dewey 방법[5]을 사용하여 단일 문서에서 모든 Element Node는 서로 다른 ID를 갖도록 하였다. Dewey ID는 DB 스키마에서 부모 자식 테이블간 연결을 맺는 포인터의 역할을 하여 분해된 문서를 쉽게 재조합할 수 있다.

TV-Anytime 메타데이터는 단일의 XML Schema를 사용하지 않고 Inlining방법은 사용하지 않았다. 대신 Path Table을 따로 두어 각각의 Node가 저장된 Table을 찾기 위한 Path정보를 기록하여 XPath[6] 표현의 질의를 처리할 경우 Path Table과의 비교를 통하여 불필요한 Join의 수를 제거 할 수 있다. 또한, 단일의 XML문서에 질의 하기 위하여 각각의 TV-Anytime 메타데이터마다 문서의 ID를 부여하고 이를 저장하고 관리하기 위한 Document Table을 추가하였다.

```
<TVAMain version="1">
  <ProgramInformation
    programId="crid://www.imbc.com/2001001">
    <BasicDescription>
      <Title>장도</Title>
      <Synopsis>복합산 봉수대에서 봉화가 피어 오르자 한양은
        천란에 대한 소문으로 술렁이고...</Synopsis>
      <Keywords>
        <mpeg7:Keyword>어울전</mpeg7:Keyword>
      </Keywords>
    </BasicDescription>
  </ProgramInformation>
  <OnDemandProgram>
    <Program crid="crid://www.imbc.com/2001001"/>
    <URL>C:\wmedia_data\Wdrama\Wsangdo.mpg</URL>
  </OnDemandProgram>
</TVAMain>
```

[그림 3] DB Table설계를 위한 TV-Anytime 메타데이터

[그림 3]은 [그림 4]와 [그림 5]의 DB Schema설계를 위한 Sample TV-Anytime 메타데이터 이고, [그림 4]는 각 Element Table과 Path Table과 Document Table들과의 관계를 나타내고 있으며 [그림 5]는 각 Element Table사이의 관계를 나타내고 있다.

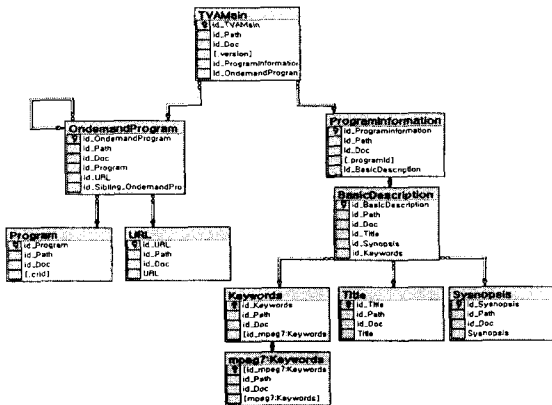


[그림 4] Element, Path, Document Table들의 관계

TV-Anytime 메타데이터의 기술에 사용되는 모든 Element Node들은 동일한 이름을 가지는 Node끼리 동일한 Table에 저장되고 모든 Element Table은 위[그림 4]와 같이 Path Table과 Document Table과 연결되는 구조로 DB Schema는 구성되어 있다. Path Table의 PathExp 필드는 XPath 질의의 처리를 위해 각 Element Node의 Path를 저장하여 검색 시간을 효과적으로 줄일 수 있다. Document Table의 CRID 필드는 TV-Anytime 메타데이터의 삭제 또는 수정 시 사용되는 TV-Anytime 메타데이터마다 부여되는 고유 키인 CRID를 저장한다.

[그림 5]는 [그림 3]의 간단한 TV-Anytime 메타데이터를 위한 DB 다이어그램으로 각 Element Node마다 하나의 Table을 생성하고 있다. Element node의 Text 또는 Attribute Node는

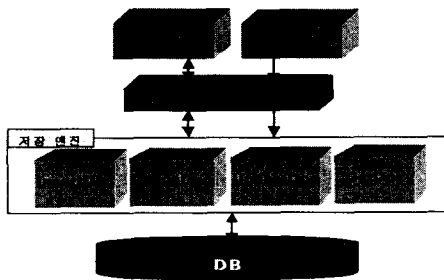
해당 Element Table의 필드로 정의된다.



[그림 5] TV-Anytime 메타데이터의 DB 다이어그램

3.2 TV-Anytime 메타데이터 저장 엔진의 구조

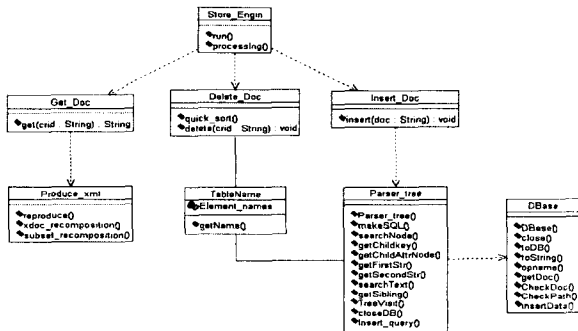
저장엔진은 [그림 6]과 같은 구조를 갖는다.



[그림 6] 저장 엔진의 구조

메타데이터 저장엔진은 메타데이터의 삽입, 삭제와 갱신을 위하여 InsertDoc, DeleteDoc, UpdateDoc, GetDoc의 네 개의 모듈로 구성되어 있다.

저장엔진은 아래와 같은 클래스들로 구현된다.



[그림 7] 저장엔진 클래스 다이어그램

Store_Engin 클래스 : 저장엔진의 주 클래스로서 Store_Engin은 XML 문서를 가져오고, 삭제하고, 저장하며 갱신하는 모듈들을 필요에 따라 호출하는 역할을 한다.

Get_Doc 클래스 : TVAnyTime문서의 고유 id인 CRID를 바탕으로 DB에 존재하는 문서의 분해된 조각들을 원문으로 복원한다. Get_Doc에서 사용하는 Produce_xml클래스는 Tree와 같은 직관적인 구조를 갖는 XML문서 저장을 위한 DB스키마들의 관계를 이용하는 재귀알고리즘을 사용한다.

Delete_Doc 클래스 : CRID를 바탕으로 DB에서 특정 XML문서를 삭제하며 이때 테이블간 우선 순위 정보는 TableNames클래스에서 참조한다.

Insert_Doc 클래스 : 새로운 XML문서를 DB에 삽입할 때 Insert_Doc클래스를 사용한다. Parser_tree클래스는 XML문서를 파싱 결과인 DOM 트리를 순회하며 각 노드의 값과 함께 path정보와 키가 되는 Dewy번호를 생성하여 이들을 해당 테이블을 찾아 삽입해나간다.

4. 결론

TV-Anytime 메타데이터는 대용량이고 단일의 스키마를 가지며 문서 형식은 XML이다. 이러한 특성을 반영할 수 있는 TV-Anytime에 종속된 저장 엔진이 필요하다.

본 논문에서는 데이터베이스를 사용하여 TV-Anytime 메타데이터 전용 저장 엔진을 설계, 구현하였으며 이를 위해 TV-Anytime 메타데이터의 특성을 고려하여 범용적인 해법보다는 디지털 방송 환경에 맞게 데이터베이스 Schema를 설계하였고 이를 바탕으로 원문의 삽입, 삭제, 갱신, 인출 모듈들을 개발하였다. 저장 엔진은 TV-Anytime 관리 시스템에서 XQuery[7]와 같은 질의처리기와 연동될 수 있다

5. 참고문헌

- [1] TV Anytime Specification Series, August 2001. (<http://www.tv-anytime.org/>)
- [2] A. Deutsch, M. Fernandez, & D. Suciu, "Storing Semistructured Data with STORED," In Proc. of the ACM SIGMOD Conf. Management of Data, June 1999.
- [3] D. Florescu & D. Kossmann, "Storing and Querying XML Data Using an RDBMS," IEEE Data Engineering Bulletin, Vol. 22, No. 3, pp. 27-34, 1999.
- [4] J. Shanmugasundaram, K. Tufte, G. He, C. Zhang, D. DeWit, & J. Naughton, "Relational Databases for Querying XML Documents: Limitations and Opportunities," Proc. 25th VLDB, pp. 302-314, Edinburg, Scotland, September 1999.
- [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] W3C, XML Path Language (XPath) 2.0, WD, November 2002. (<http://www.w3.org/TR/xpath20/>)
- [7] W3C, XQuery 1.0: An XML Query Language, WD, November, 2002. (<http://www.w3.org/TR/xquery/>)