

TV-Anytime 메타데이터 검색을 위한 XQuery 엔진

박중현¹⁾ 김병규 강지훈
충남대학교 컴퓨터학과
{ jhpark⁰, yourovin, jhkang }@cs.cnu.ac.kr

An XQuery Engine for searching TV-Anytime metadata

Jong-Hyun Park⁰, Byung-Kyu Kim, Ji-Hoon Kang
Dept. of Computer Science, Chungnam National University

요 약

TV-Anytime은 국제 표준으로 자리잡아가고 있는 차세대 디지털 방송용 메타데이터로서 사용자가 직접 원하는 방송을 검색하고 선택하여, 선택한 멀티미디어 데이터를 저장하였다가 시간과 장소에 구애받지 않고 사용할 수 있도록 하는 것을 목표로 하고 있다. 이러한 TV-Anytime 메타데이터를 효율적으로 사용하기 위해서 TV-Anytime 메타데이터를 검색하기 위한 검색엔진은 필수적이다.

본 논문에서는 TV-Anytime 메타데이터가 XML 형식을 따른다는 점을 고려하여 XML 데이터를 검색하기 위한 표준 질의어인 XQuery를 TV-Anytime 메타데이터의 검색을 위한 질의어로 사용하여 XQuery엔진을 설계하고 구현하였다. 본 논문에서 구현한 XQuery엔진은 XQuery를 검색어로 사용하기 때문에 디지털방송 환경에서 시스템 간의 상호 운용성을 높이고, 보다 풍부한 질의를 가능하도록 한다. 또한 XQuery엔진은 DB에 엘리먼트 단위로 분할되어 저장된 TV-Anytime 메타데이터를 효율적으로 검색할 수 있도록 XQuery를 SQL로 변환하는 기능을 수행한다.

1. 서론

TV-Anytime[1]은 차세대 디지털 방송용 메타데이터 국제 표준으로, 기존의 방송 시스템에서 방송 서비스 제공자가 사용자에게 방송 프로그램을 일방적으로 전송하는 것과는 달리 사용자가 직접 방송 프로그램을 검색하고 선택할 수 있으며, 사용자가 소유하고 있는 저장 장치에 프로그램을 저장하여 언제든지 필요할 때 방송 프로그램을 이용하는 것을 목표로 하고 있다. 이러한 것을 가능하게 하기 위한 수단으로서 방송 서비스용 메타데이터를 표준화한 것이 TV-Anytime이다.

TV-Anytime을 기반으로 한 방송 시스템에서 TV-Anytime 메타데이터는 XML[2]의 형식으로 표현된다. 그러므로 방송용 메타데이터의 검색은 결국 XML 문서의 검색의 문제이다. 다만 그 응용분야가 방송용이고, 그 양이 대용량이라는 특징을 가지고 있다.

본 논문에서는 TV-Anytime 문서의 검색을 위해서 XML 데이터의 검색을 위한 국제 표준 질의어인 XQuery[3]를 메타데이터의 검색을 위한 질의어로 사용하도록 제한함으로써 시스템간의 상호 운용성을 높이도록 한다. TV-Anytime과 같은 대용량의 메타데이터 저장과 관리를 위해서는 주로 데이터베이스를 사용하기 때문에 사용자 질의를 XQuery로 변환하고, 다시 XQuery를 DB에 질의 할 수 있는 SQL로 변환하는 XQuery엔진을 설계하고 구현하였다.

본 논문의 구성은 다음과 같다. 2절에는 XML문서의 검색을 위한 기존의 연구들과 TV-Anytime 메타데이터 방송 시

스템에 대해서 기술하고 있다. 3절에서는 TV-Anytime 메타데이터 검색을 위한 XQuery엔진의 설계와 구현을 기술하고 있으며, 마지막으로 4절에서 결론을 맺는다.

2. 연구배경

2.1 관련연구

XML 데이터의 검색을 위해서는 XML-QL[4], XQL[5], XQuery등과 같은 질의어가 존재하지만 현재 인터넷 표준화 단체인 IETF와 W3C는 공동으로 XML 데이터를 위한 질의어로 XQuery를 국제 표준으로 제정 중에 있다. XQL의 경우 XPath[6]의 표현을 이용하여 XML 문서에 대해 질의 하는 질의어로 단일의 XML문서에 대해서는 효율적이지만 데이터베이스에 분할 되어 저장되어 있는 XML 데이터에 대해서는 질의하기 힘들다는 단점이 있으므로 본 논문에서 제안하고 있는 TV-Anytime 메타데이터의 검색 용으로는 부적합하다고 볼 수 있다. XML-QL의 경우는 XQL과는 달리 SQL과 유사한 질의어의 구조를 따르기 때문에 데이터베이스에 저장되어 있는 XML 데이터에 대한 질의는 효율적이지만 XML 문서를 대상으로 질의를 하기 어렵다는 단점이 있다. 이러한 기존 XML 질의어들의 단점을 보완하여 XML 데이터의 저장 형식에 상관없이 XML 데이터를 검색할 수 있는 질의어가 XQuery이다.

XQuery를 XML데이터 검색을 위한 질의어로 사용하여 DB에 저장되어 있는 XML 데이터를 검색하기위한 연구로는 IBM에서 연구 개발하고 있는 XPERANTO가 있다[7,8,9]. XPERANTO 시스템은 XML 뷰 기반으로 질의 처리가 이루어지는 관계형 데이터베이스 환경에서 설계된 미들웨어 시

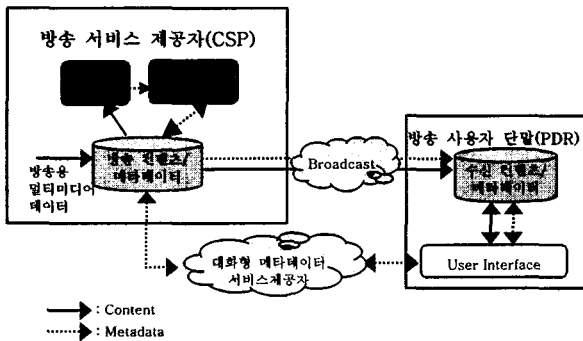
¹⁾ 이 연구는 BK21 충남대학교 정보통신 인력 양성 사업단의 지원을 받았다.

스텝으로 관계형 데이터를 XML 뷰로 정의하기 위해 XQuery를 사용하며 사용자 질의 또한 XQuery이다. 즉, DB에 저장되어있는 XML 데이터를 XQuery를 이용하여 질의 할 수 있도록 XML 뷰를 제공하고 사용자는 XML 뷰에 질의하여 검색하는 방법이다. 이러한 XPERANTO 시스템의 목표는 모든 다양한 XML 문서의 저장과 검색을 가능하게 하는 것이므로 저장과 질의의 처리 과정이 복잡할 수 밖에 없다. 하지만 본 논문에서 연구 개발 하고 있는 TV-Anytime 메타데이터는 단일의 XML 스키마를 따른다는 특성이 존재하므로 보다 특화된 방법이 필요하다.

2.2 TV-Anytime 메타데이터 관리시스템

디지털 방송은 차세대 방송의 새로운 패러다임으로서, 화질의 획기적인 개선은 물론이고 사용자 서비스 측면에서 기존의 공중파 방송에서는 불가능하였던 다양한 서비스를 할 수 있다. TV-Anytime은 이러한 차세대 디지털 방송용 메타데이터 표현의 새로운 표준으로 XML의 형식을 따르며 방송용 멀티미디어 데이터의 검색, 사용자 기록/선호도 관리, On-Demand 서비스 등의 새로운 부가 서비스를 원활하게 지원하기 위해서 필요한 메타데이터를 정의하고 있다[10,11].

TV-Anytime 메타데이터 기반의 방송 시스템은 현재 한국 전자통신 연구원에서 개발 중에 있으며 [그림 1]은 TV-Anytime 메타데이터 기반 방송서비스 시스템의 기능 구성도이다. 메타데이터 기반 방송서비스 시스템은 크게 방송 서비스 제공자(Content Service Provider)와 서비스를 사용하는 방송 사용자 단말(Personal Digital Recorder)로 구성되고, 방송 사용자 단말은 다시 모바일 폰과 같은 하위의 방송 사용자 단말(sub-PDR)로의 데이터 전송이 가능하다.



[그림 1] 메타데이터 기반 방송 서비스 시스템

본 논문에서 설계하고 구현한 XQuery엔진은 CSP에서 메타데이터의 검색을 위해 사용 할 수 있도록 하고있으며, 향후 PDR에서 검색을 요청 할 경우 사용 될 수 있을 것으로 판단된다.

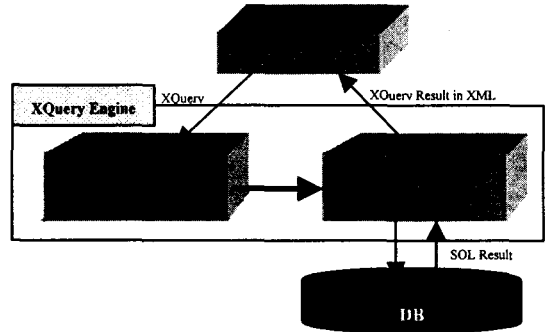
3. XQuery엔진의 설계 및 구현

3.1 XQuery엔진의 구조

본 논문에서는 TV-Anytime 메타데이터를 위한 스키마를 분석하여 간선단위로 동일한 이름을 가지는 모든 엘리먼트 Node는 동일한 테이블에 저장하는 방법으로 DB 스키마를 정의했다. 이러한 방법은 데이터베이스의 저장 효율을 높일 수 있고, 확장이 용이하다는 장점을 가지는 대신, XQuery에 기술되어 있는 XPath 표현을 처리하기 위해서는 많은 join이 필요하다는 단점이 존재한다[12]. 이러한 단점을 보완하기

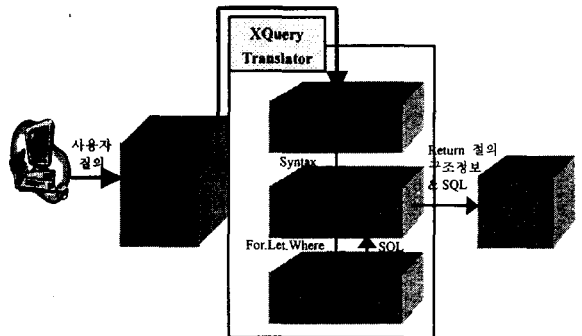
위해서 본 논문에서는 Root Node에서 해당 엘리먼트 Node까지의 전체경로와 해당 엘리먼트 Node의 정보가 저장되어 있는 테이블의 ID를 속성으로 가지는 Path 테이블을 추가하여 불필요한 Join의 수를 줄임으로써 단일의 스키마를 따르는 TV-Anytime 메타데이터에 특화된 DB 스키마를 제안한다.

XQuery 엔진의 구조는 [그림 2]와 같다. XQuery엔진은 방송서비스 제공자 시스템의 하나의 모듈로서 존재하며, 방송서비스 제공자는 메타데이터의 검색을 위하여 메타데이터 검색기를 사용한다. 주어진 질의는 XQuery 질의로 변환이 되며, XQuery 질의는 XQuery 엔진으로 보내어지며, 엔진은 질의를 처리하여 그 결과를 XML 문서의 형태로 메타데이터 검색기로 반환한다.



[그림 2] XQuery 엔진의 구조

XQuery Translator는 XQuery 엔진의 모듈로서 사용자가 질의한 XQuery를 입력으로 하여 DB에 질의하기 위한 적절한 SQL로 변환하는 기능과 XQuery에 기술되어 있는 반환될 XML문서의 구조정보를 얻는 작업을 수행하는 모듈로 [그림 3]과 같은 구조를 가진다.

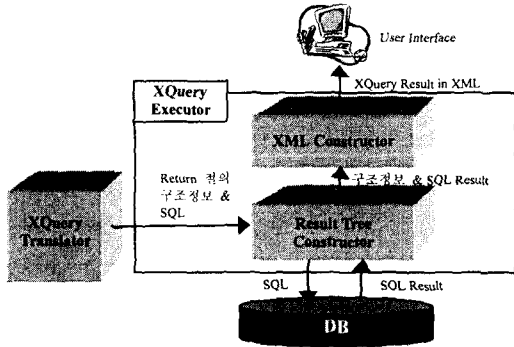


[그림 3] XQuery Translator의 구조

사용자가 입력한 XQuery는 XQuery Parser를 통하여 구문분석한 XQuery Syntax Tree를 얻어낸다. 질의 분석기는 XQuery Syntax Tree에서 FOR, LET, WHERE와 RETURN절에 저장된 정보들을 얻어내어 FOR, LET, WHERE절에서 얻어낸 정보는 SQL로 변환하기 위하여 XQuery2SQL로 보내지고 RETURN절에 포함되어 있는 정보는 반환될 XML문서의 구조정보를 포함하고 있으므로 반환할 XML의 구조정보를 추출한다. XQuery2SQL은 질의분석기에서 입력 받은 정보를 가지고 적절한 SQL로 변환하고 변환된 SQL을 다시 질의 분석기로 반환한다. 질의분석기는 SQL과 사용자에게 반환하여야 할

XML 구조정보를 XQuery Executor에게 제공한다.

XQuery Executor의 구조는 [그림 4]와 같은 구조로 구성된다.



[그림 4] XQuery Executor의 구조

Result Tree Constructor는 XQuery Translator로부터 반환한 XML문서의 구조정보와 DB에 질의하기 위한 SQL을 입력으로 받아 실제 메타데이터가 저장되어 있는 DB에 질의를 수행하여 SQL 결과 집합을 얻는다. 이때 SQL의 SELECT절에 정의된 값이 메타데이터의 중간 Node의 반환을 요청하고 있다면 해당 테이블의 ID를 반환할 것이고, 그렇지 않다면 반환을 요구하는 Node는 말단 Node 또는 Attribute Node이므로 메타데이터에 실제 들어있는 내용을 반환할 것이다. 만약, 반환된 값이 테이블의 ID라면 XML Constructor는 Result Tree Constructor에게 현재 테이블에 해당하는 Node의 모든 자식 Node를 반환하도록 요청하여 하위의 모든 자식 Node들을 포함하는 XML문서를 생성하여 사용자에게 반환한다.

3.2 XQueryToSQL 변환 알고리즘

다음은 XQuery를 SQL로 변환하기 위한 알고리즘이다.

XQuery의 Where절 처리 알고리즘

1. XQuery의 Where절의 변수를 For절에서 구한다.
2. For절의 Path표현 중 가장 마지막 엘리먼트 Node의 이름은 DB의 테이블의 이름이 된다.
3. 사용자가 검색을 요청한 질의어와 동일한 값을 검색.
 - (1) Where절에 해당하는 변수가 엘리먼트 Node를 표현하고 있으면 질의해야 할 필드의 이름은 테이블 이름과 동일.
 - (2) XQuery의 Where절에 해당하는 변수가 Attribute Node를 표현하고 있으면 질의해야 할 필드의 이름은 Attribute의 이름.
4. Where절의 변수에 해당하는 Full Path 표현을 XQuery의 모든 For절을 검색하여 얻어낸다.
5. Path 테이블에 저장되어있는 Full Path값과 동일한지 질의
6. Path 테이블의 Full Path값에 해당하는 Node를 저장하고 있는 테이블과 2에서 얻은 테이블과 동일한지 질의.

XQuery의 Return절 처리 알고리즘

1. XQuery의 Return절의 변수를 For절에서 구한다
2. For절의 Path표현 중 가장 마지막 엘리먼트 Node의 이름은 DB의 테이블의 이름이 된다.
3. DB에 질의하기 위한 테이블의 필드 이름을 선택.
 - (1) Return절에 해당하는 변수가 말단 엘리먼트 Node를 표현하면 필드의 이름은 테이블의 이름과 동일.
 - (2) Return절에 해당하는 변수가 말단이 아닌 엘리먼트를 표현하고 있으면 질의해야 할 필드의 이름은 테이블의 ID.
 - (3) Return절에 해당하는 변수가 Attribute Node를 표현하고 있으면 질의해야 할 필드의 이름은 Attribute의 이름과 동일.

4. 구조 검색을 위해 SQL의 WHERE절에 조건을 추가.

- (1) 반환되는 값을 가지는 테이블과 조건을 만족하는 테이블이 동일한 문서 ID를 가지는지 질의.
- (2) XQuery의 Return절 변수와 Where절 변수가 동일한 Path를 포함하고 있으면 같아지는 엘리먼트 Node에 해당하는 테이블 ID가 Return절 변수에 해당하는 테이블의 ID와 Where절 변수에 해당하는 테이블의 ID에 모두 포함되는지 질의.

변환된 SQL을 DB에 질의하여 검색한 결과가 말단의 엘리먼트 Node, 또는 Attribute Node 이면 질의의 결과는 해당 Node의 내용이 되므로 해당 필드의 이름을 Node의 이름으로 하고, 검색된 값을 Node의 값으로 하는 단일 Node를 생성하여 반환한다. 만약 검색한 결과가 자손을 가지는 Node 라면, SQL의 결과 값은 해당 Node의 ID가 반환되므로 자식 Node와의 Join을 통해 말단 Node까지의 sub-Tree를 반환한다.

4. 결론

XML 형식을 따르는 TV-Anytime 메타데이터의 검색을 위한 질의어로 XQuery를 사용하는 것은 상호 운용성 면에서 매우 효과적이다[13]. 사용자는 메타데이터의 저장 형식에 무관하게 질의 할 수 있고, 각 응용간에도 상호 데이터 검색을 위한 질의어로 사용 가능하다. 본 논문에서 개발한 XQuery 엔진은 분산된 TV-Anytime 메타데이터의 검색을 위해 사용자 질의를 XQuery로 변환하고, 대용량의 데이터를 저장 하기 위해 가장 보편적으로 사용되는 데이터 베이스 질의어인 SQL로 변환하는 기능을 수행함으로써 TV-Anytime 기반 방송시스템에서 검색을 위해서 사용된다. 향후, XQuery 엔진은 TV-Anytime기반의 방송 시스템에서 뿐만 아니라 XML 데이터의 검색을 필요로 하는 여러 가지 응용에서도 사용 가능 할 것이다.

5. 참고문헌

- [1] TV Anytime Specification Series, August 2001.
- [2] W3C, Extensible Markup Language (XML) Version 1.0, Recommendation, February 1998.
- [3] W3C, XQuery 1.0: An XML Query Language, Working Draft, November, 2002.
- [4] W3C, XML-QL: A Query Language for XML, Note, August 1998.
- [5] W3C, XML Query Language (XQL), September 1998.
- [6] W3C, XML Path Language (XPath) 2.0, Working Draft, November 2002.
- [7] J. Shanmugasundaram, J. Kiernan, E. Shekita, C. Fan, & J. Funderburk, "Querying XML Views of Relational Data," Proc. 27th VLDB, Roma, Italy, September 2001.
- [8] 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.
- [9] J. Shanmugasundaram, "XPERANTO: Bridging Relational Technology and XML", IBM Research Report, June 2001.
- [10] S. Pfeiffer & U. Srinivasan, "TV Anytime as an application scenario for MPEG-7," Proc. ACM Multimedia 2000, Los Angeles, October 2000.
- [11] 한국정보통신기술협회, "TV-Anytime Forum 최근동향", Proc. IT Forum Korea 2002, May 2002.
- [12] D. Florescu & D. Kossmann, "Storing and Querying XML Data Using an RDBMS," IEEE Data Engineering Bulletin, Vol. 22, No. 3, pp. 27-34, 1999.
- [13] 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, June 2002.