내장형 XML 데이타베이스를 위한 타입 처리기의 설계 및 구현

임우규[0] 권준호 이석호
서울대학교 전기 컴퓨터공학부
{wgihm[0], bluerain}@db.snu.ac.kr shlee@cse.snu.ac.kr

Design and Implementation of the Type Processor in the Embedded XML Database System

Woogyu Lim[0] Junho Kwon Sukho Lee
School of Electrical Engineering and Computer Science, Seoul National University

요 약

문서를 표현하는 여러 가지 기법 중에서 XML(eXtensible Markup Language)은 확장성과 범용성 때문에 데이타의 교환과 표현에 대한 표준으로 자리잡고 있다. 이에 XML로 데이타를 저장하기 위한 다양한 방법들이 제시되고 있다. 본 논문에서는 관계형 데이타베이스에 XML 데이타를 저장하기 위해 XML 데이타의 타입을 관계형 데이타베이스의 타입으로 매칭해 주는 서브시스템인 타입 처리기를 구현한다. 타입 처리기에서는 eXDM이 내장형 시스템이라는 제한사항을 고려하고, XQuery 형태의 질의문을 지원하기 위해 XML-스키마를 변형한 형태의 XML 타입 정의 문서를 이용한다.

1. 서 론

XML[1]은 데이타 표현의 유연성으로 인해 인터넷 상에서 주고 받는 데이타의 표준으로 자리 잡고 있으며, TV-Anytime[2], MPEG-7, ebXML, 웹서비스 등의 최신 기술에서 널리 사용되고 있다. Oracle 9i, Tamino[3]와 같은 데이타베이스 시스템에서도 XML 데이타의 저장과 검색을 지원하고 있으며, 관련 응용 시스템에 대한 개발도 지속적으로 이루어지고 있다.

eXDM(embedded XML Data Management System)[4][5]은 관계형 데이타베이스 시스템을 기반으로 하는 내장형 XML 저장 및 검색 시스템이다. 질의 언어로는 XQuery를 사용하며 내부적으로 XQuery를 SQL로 변환하여 질의를 수행한다. 그리고 내장형 시스템이기 때문에 전체 시스템의 크기에 제한이 있다는 점에 중점을 두고 시스템을 설계하였다.

본 논문에서는 관계형 데이타베이스에서 XML 데이타를 저장하고, 검색하기 위해서 반드시 수반되어야 하는 타입 처리 서브 시스템을 구현하였다. 타입 처리 서브 시스템에서는 XML-스키마[6]를 변형한 형태의 XML 타입 정의 문서를 이용하여 정확한 타입 정보를 제공하고, 내장형 시스템을 위한 시스템의 크기를 줄이는 기법을 제안하였다.

본 논문의 구성은 다음과 같다. 2절에서 관련 연구에 대한 내용을 다루고, 3절에서는 eXDM에 대한 전체 구조를 살펴본다. 그리고 4절에서는 타입 처리기의 구조와 처리 과정에 대해 살펴보고, 5절에서 결론을 맺는다.

2. 관 련 연 구

XML 데이타를 관계형 데이타베이스 시스템에 저장하기 위해서 구조 정보와 타입 정보의 제공이 반드시 필요하게 되는데[7], 이러한 것으로 대표적인 기법이 DTD(Document Type Definitions)이다. 그러나 DTD는 XML의 형식을 따르지 않고 있어서 처리가 복잡하며, 타입 정보의 제공도 제한적이다. XDuce[8]와 같은 시스템에서는 XML 타입을 처리하기 위해서 별도의 타입 정의 언어를 제공하여 타입 정보를 처리하고 있다. 하지만 이러한 방법 역시 XML의 형식과 다른 형식을 이용한다는 점에서 부가적인 작업이 필요하게 되고, 이는 전체 시스템의 크기를 커지게 할 수 있다는 점에서 내장형 시스템에는 적합하지 않다. 최근에 제안된 XML-스키마 기법은 XML 형식의 구조 정의 문서로 데이타의 타입 뿐만 아니라 그 구조까지 한번에 정의할 수 있는 장점을 가지고 있다. eXDM에서는 내장형 시스템에 적합하도록 XML-스키마를 변형한 XML 타입 정의 문서를 이용하고 있다.

3. 내장형 XML 데이타베이스 시스템을 위한 타입 처리기

3.1. 전체 시스템의 구조

XML 데이타를 저장 및 검색하기 위한 eXDM은 크게 XML 타입 처리기, XML 데이타 로더, XML 인덱스 처리기, XQuery 프로세서 그리고 내장형 SQL 엔진으로 구성된다. XML 데이타 로더는 XML 데이타 문서를 번호 부여 기법(Numbering Scheme)[9]을 이용하여 문서를 파싱한 후, 관계형 데이타베이스의 테이블에 저장한다. XML 인덱스 처리기는 빠른 질의 처리를 위해 인덱스를 지원한다. 자주 접근이 필요한 데이타에 대한 인덱스를 구성하여 전체 데이타에 접근 없이, 새롭게 생성된 작은 크기의 인덱스 테이블만을 접근하여 검색을 할 수 있게 한다. 이는 트위그(twig) 형태의 질의에 대하여 더 좋은 성능을 제공하게 된다. XQuery 프로세서는 사용자로부터 XQuery를 입력 받아 이를 SQL로 변환하여 처리한 후, 그 결과를

1. 이 논문은 2003년도 두뇌한국21사업과 정보통신부의 대학 IT연구센터(ITRC) 지원을 받아 수행되었음

사용자에게 반환하는 역할을 한다. 사용자에게 결과를 반환하는 경우에 데이타 값만 반환할 수도 있고, XML 형태로 반환할 수도 있게 설계되었다. eXDM 시스템의 전체적인 구조는 그림 1과 같다.

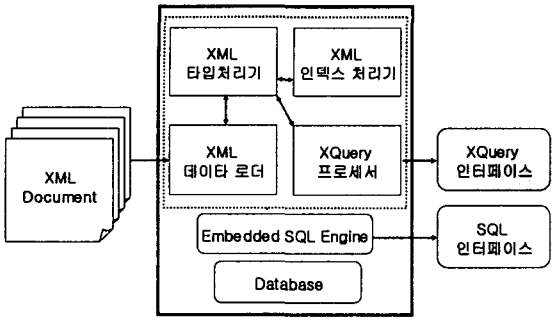

그림 1. eXDM 시스템의 구조

3.2. 타입 처리기 서브 시스템

전체 eXDM 시스템의 각 서브 시스템인 XML 데이타 로더, XML 인덱스 처리기 그리고 XQuery 프로세서는 내부적인 기능을 수행하기 위해 반드시 타입 정보를 필요로 한다. 타입 처리기는 주어진 XPath에 대해 타입 정의 문서에서 정의한 타입을 반환한다. 각 서브 시스템들은 타입 정보가 필요할 경우에 타입 처리기를 호출하여 타입 처리기가 반환해주는 타입 정보에 따라 XML 데이타를 저장하거나 검색하게 된다. 따라서 타입 처리기는 XML과 관계형 데이타베이스 간의 데이타 타입을 매칭해 주는 기능을 지원해야 한다. 그리고 다른 서브시스템에게 타입 정보를 제공하는 인터페이스를 제공하여야 한다.

4. 타입 처리기 서브시스템의 설계 및 처리과정

4.1. 구조

eXDM에서는 타입 정보를 처리하기 위해, XML 스키마를 변형한 타입 정의 문서를 사용한다. 원본 XML 문서의 엘리먼트와 애트리뷰트는 모두 XML 타입 정의 문서에서 동일한 부모, 자식의 구조를 유지한 엘리먼트로 선언한다. XML 타입 정의 문서에서 이의 구별은 node_type이라는 애트리뷰트로 지정한다. 원본 XML 문서의 엘리먼트에 대해서는 node_type을 node로 선언하고, 애트리뷰트에 대해서는 attribute로 선언한다. 원본 XML 문서에서 데이타를 갖는 엘리먼트와 애트리뷰트의 데이타 타입을 위해서 XML 타입 정의 문서는 data_type을 이용한다. int, string 등과 같이 매칭되어야 하는 타입 정보는 data_type 애트리뷰트로 선언되어야 한다. 그림 3은 그림 2의 XML 문서에 대한 타입 정의 문서이다.

```
<TVAMain version="0.0"
         publisher="Seoul National University"
         publicationTime="2002-08-21T14:27:10 ">
  <ProgramDescription>
    <Synopsis>Long Text Here</Synopsis>
  </ProgramDescription>
  <ProgramDescription>
    <Synopsis>Another Long Text Here</Synopsis>
  </ProgramDescription>
  ......
</TVAMain>
```

그림 2. 원본 XML 문서

```
<TVAMain data_type="internal">
  <version node_type="attribute"
           data_type="float"/>
  <publisher node_type="attribute"/>
  <publicationTime node_type="attribute"
                  data_type="datetime"/>
  <ProgramDescription data_type="internal">
    <Synopsis data_type="text"/>
  </ProgramDescription>
  ......
</TVAMain>
```

그림 3. XML 타입 정의 문서

예를들어, 'TVAMain'은 엘리먼트이므로 node_type이 node가 된다. eXDM에서는 node 타입을 디폴트로 사용하므로 예제에서는 node_type이 생략되어 있다. 'version'은 원본 XML 문서에서 'TVAMain'의 애트리뷰트에 해당된다. 따라서 타입 정의 문서에서는 'version'을 'TVAMain'의 하위 엘리먼트로 선언하고, node_type은 attribute로 지정하며, data_type은 실수형 데이타를 갖는 float 타입이 된다. 'publisher'의 경우는 디폴트인 string이므로 data_type에 대한 정의가 생략되어 있으며, 'Synopsis'는 SQL에서 255바이트 크기인 varchar 타입보다 큰 string을 지원하기 위하여 data_type을 text 타입으로 선언한다.

다음의 표 1과 표 2는 타입 정의 문서에서 사용되는 node_type과 data_type을 나타내고 있다. 이 표에서는 eXDM 시스템에서 요구하는 XML의 타입을 정의하고 있다. node 타입은 XML 문서에서 엘리먼트에 해당된다. node 타입 중에서도 특별히 자식으로 텍스트 데이타를 갖지 않고, 엘리먼트와 애트리뷰트만을 자식으로 가지는 엘리먼트의 데이타 타입은 internal 타입으로 정의하고 있다. 이는 eXDM 시스템이 번호 부여 기법(Numbering Scheme)에 기반하여 설계되어 있기 때문에 필요없는 엘리먼트들이 테이블에 저장되는 것을 방지하여 성능을 향상 시키기 위함이다. 또한 eXDM은 날짜에 대한 정보가 중요한 의미를 갖는 TV-anytime 시스템을 위하여 설계되었으므로 날짜 정보에 관한 데이타 타입을 datetime과 duration으로 분리하여 지원하고 있다.

표 1. node_type의 종류

node_type의 종류
node (디폴트)
attribute

표 2. data_type의 종류와 EXDM 타입매칭

data_type	EXDM_TYPE
int	EXDM_INTEGER
float	EXMD_FLOAT
string (디폴트)	EXDM_STRING
text	EXDM_TEXT
datetime	EXDM_DATETIME
duration	EXDM_DURATION
internal	EXDM_INTERNAL

4.2. 구현

타입 처리기를 구현하기 위하여 그림 3과 같이 미리 정의한 XML 타입 정의 문서를 DOM(Document Object Model)으로 파싱한다. XML4J 라이브러리를 이용하여 파싱을 하면, 메모리에

DOM 트리가 생성된다. 일반적으로 XML 타입 정의 문서는 크기가 원본 XML 문서보다 훨씬 더 작고, 다른 서브 시스템이 자주 이용을 한다는 특징을 갖는다. 따라서 SAX(Simple API for XML)을 이용하는 것보다 DOM을 이용하는 것이 좋은 성능을 갖는다. 메모리에 트리가 생성되면, 주어지는 XPath를 바탕으로 트리를 순회하여 타입 정보를 반환하는 것이 가능하다. XPath는 완전한 경로로 주어질 수도 있지만, '//Programinformation' 과 같이 간단하게 주어질 수도 있으므로 트리를 순회하는 일련의 과정은 FA(Finite Automata) 검색을 지원해야 한다. 타입 처리기는 이러한 것을 TH_GetElementType 모듈에서 구현하고 있다. 이 모듈은 다른 서브 시스템으로부터 XPath를 입력으로 받아서 타입 정보를 반환하는 역할을 한다.

4.3. 기 능

타입 처리기의 기능을 전체적으로 나타내면 다음의 그림 4와 같다. 다른 서브 시스템에서 데이타의 저장과 검색을 위하여 반드시 타입 정보가 필요한데, 타입 처리기가 메모리에 상주하여 각 서브시스템이 호출 할 때마다 타입 정보를 제공한다.

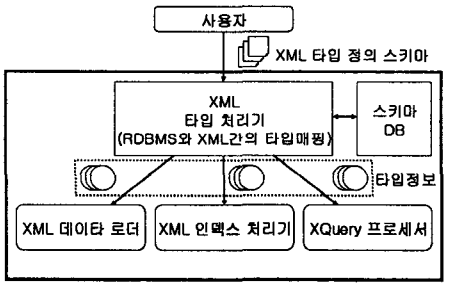

그림 4. 타입 처리기와 다른 서브시스템과의 관계

데이타 로더는 원본 XML 문서를 구조적으로 분석하여 관계형 데이타베이스에 저장하게 된다. eXDM은 관계형 데이타베이스에 저장하기 위하여 번호 부여 기법(Numbering Scheme)을 이용하기 때문에, 타입별로 value 테이블이 따로 생성되는 구조를 갖고 있다. 따라서 어떤 타입의 테이블에 저장을 할 것인지에 대한 정보를 알아야 적합한 테이블에 저장할 수가 있게 되는데, 이 정보를 타입 처리기가 제공한다. 예를 들어, 데이타 로더는 그림 2의 XML 문서를 파싱할 때, 'TVAMain'엘리먼트 다음의 'version'애트리뷰트에 대해서 '/TVAMain/@version'과 같은 XPath를 얻어낼 수 있다. 이러한 XPath를 타입 처리기로 넘겨주면, 타입 처리기는 이미 생성되어 있는 타입 트리를 순회하여 'EXDM_FLOAT'라는 타입을 반환한다. 이를 통해 데이타 로더는 반환된 타입정보인 'EXDM_FLOAT'를 바탕으로 관계형 데이타베이스의 테이블 중에서 필드의 데이타 타입이 float인 테이블에 'version'의 값을 저장한다.

XQuery 프로세서는 사용자로부터 주어진 XQuery를 관계형데이타베이스에서 검색하기 위하여 SQL로 변환한다. 변환된 SQL로 질의를 수행하고, 검색된 결과를 텍스트나 XML 형태로 사용자에게 반환한다. XQuery 프로세서는 사용자가 그림 5와 같은 XQuery로 질의를 할 때 이를 처리하게 된다. 예를 들어 그림 5와 같이 사용자로부터 XQuery 형태로 질의가 주어졌을 때,

```
FOR $pi IN //ProgramInformation, $be IN //BroadcastEvent
WHERE (($be//PublishedTime>'2002-08-21T19:00:00')
  OR ($be//PublishedTime>'2002-08-22T19:00:00'))
  AND $pi//Genre/Name='드라마'
  AND $pi//@programId = $be//@crid
RETURN $pi//Title/text(), $be//PublishedTime/text()
```

그림 5. XQuery 형태의 질의문

'$be//PublishedTime', '$pi//Genre/Name'과 같은 XPath가 포함되어 있는데, 이러한 XPath를 타입처리기한테 넘겨주어 어떤 테이블에서 '2002-08-21T19:00:00', '드라마'와 같은 데이타 값을 찾을지를 그 결과를 통해 판단하게 된다. 그림 5의 예에서는 주어진 XQeury 형태의 질의문을 필드의 데이타 타입이 datetime과 string인 테이블에서 조건에 맞는 투플들을 찾는 SQL 형태의 질의문으로 변환이 가능하다.

질의 처리를 빠르게 하기 위해서 eXDM은 인덱스를 지원하고 있다. 필요한 투플들만을 지정하여 작은 테이블인 인덱스 테이블에 저장한다. 이로서 자주 접근이 필요한 데이타에 대하여 효율적인 검색이 가능하다. 인덱스의 생성시에도 역시 타입 처리기를 이용하여 해당 테이블에서 필요한 투플만을 따로 인덱스 테이블에 저장하는 것이 가능하다. 이는 같은 타입을 갖는 데이타가 많을 때 훨씬 뛰어난 성능을 갖게 된다.

5. 결 론

본 논문에서는 관계형 데이타베이스 시스템을 이용하여 XML 문서를 저장하고 검색할 때 필요한 타입 처리 서브시스템을 구현하였다. 전체 시스템의 크기 제한이 있는 경우를 최대한 고려하여 서브 시스템을 구현하였고, XML의 저장이 번호 부여 기법을 이용하여 이루어지는 특성을 고려하여 디자인되고 구현되었다.

향후 과제로서 내장형 시스템이 아닌 환경에서 더 많은 기능을 제공하기 위하여 XML-스키마를 지원하도록 개선할 수 있을 것이다.

참고 문헌

[1] Tim Bray, Jean Paoli, C.M. Sperberg-McQueen and EveMaler, "Extensible Markup Language(XML) 1.0 second edition W3C recommendation." Technical Report REC-xml-20001006, World Wide Web Consortium, October 2000.

[2] TV Anytime Forum, http://www.tv-anytime.org.

[3] H. Schoning, "Tamino – A DBMS designed for XML", Proc. Of the 17th ICDE conference, April 2001.

[4] 권준호 외 6명, "내장형 XML 저장 및 검색 시스템의 구현", 한국정보과학회 춘계 학술발표논문집, Vol.30, No.1, pp.581-583, 2003.

[5] Joonho Kwon, et al., "Development of Embedded System for storing and retrieving XML data", The 15th Conference on Advanced Information Systems and Engineering(CAiSE '03), Klagenfurt/Velden, Austria, June 16-20, 2003.

[6] XML Schema, http://www.w3.org/XML/Schema

[7] Dan Suciu, "The XML Typechecking Problem", SIGMOD Record Vol. 31 No.1, March 2002.

[8] Haruo Hosoya, Benjamin C. Pierce, "XDuce: A Typed XML Processing Language(Preliminary Report)", In International Workshop on the Web and Databases, vol. 1997 of Lecture Notes in Computer Science, 2000.

[9] Q. Li and Bongki Moon, "Indexing and Querying XML Data for Regular Path Expressions", Proc. Of the 27th VLDB Conference, 261-370, September 2001.