2001년 한국정보처리학회 춘계 학술발표논문집 제8권 제1호

응용 프로그램을 위한 일관된 XML 뷰 제공에 관한 연구

남철기*, 김혜경**, 배재학*
*울산대학교 컴퓨터 정보통신공학부
**울산대학교 정보통신대학원
e-mail : cholki.nam@oracle.com

A Study on Giving a Consistent XML View to Applications

Chul-Ki Nam*, Hye-Kyung Kim**, Jae-Hak J. Bae*
*School of Computer Engineering and Information Technology
**Graduate School of Information and Communication Technology,
University of Ulsan

요 약

점차로 이종의 데이터와 문서가 XML 로 통합되는 것이 요구된다. 현재 데이타베이스 기반의 응용 프로그램들은 데이터들이 대부분 테이블에 저장되어 있기 때문에 SQL 을 이용하여 데이터처리를 하고 있다. 한편 현 정보기술 발전 추세는 응용 프로그램에서 처리하고자 하는 단위가 데이터(data-centric)에서 점차 문서(document-centric)로 바뀌어 가고 있다. XML 은 이런 패러다임에에 적합한 해결책이 될 수 있다. 본고에서는 일반적인 데이터에 대한 뷰(view)를 테이블 뷰가 아닌 XML 뷰를 응용 프로그램 개발자에게 제시하는 방법론을 보임으로써, 일관되고 유연한 문서처리 시스템 설계를 하는데 있어 대단히 유용한 또다른 한 방법론을 제공하고자 한다.

1. 서론

XML(eXtensible Markup Language)은 과거 전자문서의 논리적 정보와 물리적 정보가 같이 표현되던 구조에서 문서의 논리적구조(XML), 물리적 구조(XSL,CSS), 문서의 연결(Xlink,Xpointer)를 분리하려는 시도와 요구에 의해 생겨나게 되었다. XML 은 이상적인 전자문서의 요건을 갖추고 있으며 현재 모든 형태의 데이터와 문서를 통합, 저장, 처리할 수 있는 프레임웍을 제공한다. 이처럼 점점 문서의 표준으로 자리를 잡아가는 XML 을 응용 프로그램 환경에서 개발자들에게 일관된 방법으로 저장시스템에 저장되어 있는 데이터 또는 문서를 접근할 수 있는 방안을 제시 하고자 본 고에서는 현재 널리 사용되고 있는 객체-관계형 데이터베이스인 *Oracle8i* 를 이용하여 실험하였다.

구성은 다음과 같다. 2 절에서는 XML 과 데이터베이스에 대해서 설명하고 3 절에서는 저장시스템에 XML 문서를 저장하는 여러가지 방법에 대해서 설명한다. 4 절에서는 실제 XML 뷰를 제공하기 위해 XML 문서를 데이터베이스에 저장하고 데이터베이스 스키마와 테이블 내용을 XML 문서로 추출하는 방법에 대해 설명한다.

2. XML 과 DBMS

XML 과 데이터베이스와의 관계를 살펴볼 때 XML 이 XML 문서 자체만 의미 할때는 XML 은 데이터베이스가 아니다. 하지만 XML 이 XML 문서를 포함한 관련된 기술과 XML 도구를 의미할 때는 XML 도 데이터베이스라고 할 수 있다. 왜냐하면 기존의 데이터베이스에서 제공되는 저장소역할을 하는 XML 문서, 스키마(Schema)를 지원하기 위한 DTD(Document Type

Difinition), XML 스키마 언어, 질의언어인 XQL,XML-QL, 그리고 프로그래밍 인터페이스인 SAX(Simple API for XML) , DOM(Document Object Model)을 지원한다. 하지만 기존 데이터베이스에서 지원되는 대용량 저장소, 인덱스, 보안, 트랜잭션과 데이터 무결성, 트리거, 다수의 동시 사용자 지원과 관련해서는 데이터베이스라고 간주되기에는 부족하다[1].

이처럼 전자문서의 새로운 요구 형태인 XML 을 효과적으로 이용하기 위해 가장 기본이 되는 것은 저장시스템을 어떠한 구조로 할 것인가 하는 점이다. 안정성, 보안, 동시성을 중요시 한다면 파일 시스템을 XML 의 저장시스템으로 고려하지는 않을 것이다. 그렇다면 크게 관계형 데이터베이스, 객체지향형 데이터베이스, 그리고 XML 전용 저장시스템으로 나눌 수 있다. 위 3 가지 시스템은 모두 장,단점을 가지고 있으며 <표 1>과 같이 크게 구분할 수 있다[2,3,4].

<표 1>데이터베이스 비교

데이터베이스	장점	단점
관계형	DB 시장의 높은 점유율과 기존 시스템과 통합이 용이	XML 의 반구조적인 데이터 모델을 구조적인 관계형 모델로 매핑할 때 발생되는 데이터 손실이 발생할 수도 있음.
객체 지향형	데이터 모델의 유사성으로 데이터 손실없이 XML 문서 저장 가능	대용량의 데이터에 대한 복잡한 질의를 처리하는 능력이 성숙되지 않음
XML 전용	XML 의 데이터 모델을 잘 지원함	시스템의 안정성 및 구현 완료 가능성이 떨어짐

무엇보다도 XML 응용 목적에 가장 잘 부합되는 시스템이 선택되어야 할 것이다. 본 고에서 제시하는 시스템 구현을 위해 관계형 데이터베이스와 객체지향형 데이터베이스의 장점을 적용할 수 있는 객체관계형 데이터베이스인 Oracle8*i* Release3 를 이용하였다.

3. XML 문서 저장방법

XML 문서는 데이터중심(Data-centric)과 문서중심(Document-centric)으로 크게 두 가지의 범주로 나눌수 있다[5]. 데이터중심인 문서는 XML 문서가 데이터전송의 목적으로 사용되고 물리적 구조는 중요하지 않다. 문서중심인 XML 문서는 반구조적인 구조를 가지고 있고 내용은 섞여 있으며 물리적 구조가 중요하다.

저장 방법은 크게 3 가지로 나눌 수가 있고 첫번째는 문서중심인 XML 문서를 전체의 내용을 하나의 BLOB(Binary Large Object) 또는 CLOB(Character Large Object) 칼럼에 저장하는 방법이다. 이 방법은 구현은 간단한 반면 XML 문서에서 원하는 정보를 추출하기가 어렵고 문서내용의 잦은 수정을 필요로 하는 시스템에서는 적당하지가 않다. 두번째는 분할 저장기법으로 데이터중심인 XML 문서의 정보를 분할해서 하나 또는 하나이상의 테이블에 저장하는 방법이다. 즉, XML 문서의 요소(element)는 테이블의 칼럼과 맵핑된다. 마지막으로 세번째는 첫번째, 두번째방법을 합한 방법이다. 책을 예로 들면 수정이 필요하고 구간검색이 필요한 제목, 가격, 출판사 정보는 각각 하나의 테이블 칼럼에 저장하고 거의 변경이 없는 책에 대한 메타데이타인 설명정보(출판일자,저자)는 CLOB 에 저장한다. 본 고에서는 두번째 접근 방법을 사용하였다. 현재까지 문서저장 방법에 대해 제안한 논문들은 대부분 위의 첫번째, 두번째 저장방법에 대해 제안하고 있다[3,4]. 하지만 세번째 방법은 첫번째, 두번째 방법을 절충한 효과적인 저장방법이 될 수 있다.

4. 시스템 구조

많은 양의 XML 데이터를 효과적으로 저장하고 관리하기 위한 요구는 점점 증가하고 있다. 아래 그림 1 은 제안하고자 하는 시스템구조이다.

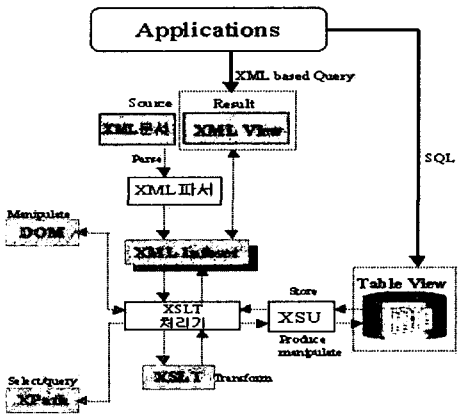

그림 1. 시스템 구조

현재 데이타베이스 기반의 응용 프로그램들은 데이터들이 대부분 테이블에 저장되어 있기 때문에 SQL 을 이용하여 테이블 데이터 처리를 하고 있다. 또한 응용 프로그램에서 처리하고자 하는 단위가 데이터(data-centric)에서 점차 문서(document-centric)로 바뀌어 가고 있다. XML 은 이런 흐름에 적합한 대안이 될 수 있다. 궁극적으로 본 고에서는 일관된 XML 뷰(View)를 응용 프로그램 개발자에게 제공하고자 한다.

XML 뷰에는 응용 프로그램 개발자가 필요로 하는 테이블의 구조(칼럼이름, 데이터 타입, 제약조건, 기본키, 외래키,...)와 인덱스정보등, 즉 데이터베이스 스키마가 XML 문서로 표현되고 필요에 따라 데이터도 포함된다. 따라서 개발자는 XML 뷰를 통해 XML 저장시스템의 종류에 관계없이 투명하게 XML 기반 질의를 하면 XML 질의어를 실제 저장시스템에서 사용되는 질의언어(SQL)로 변환되어 수행되어 데이터 처리를

할 수가 있다. 따라서 개발자 또는 데이터베이스 사용자는 그림 1 시스템 구조에서 처럼 테이블 뷰가 아닌 XML 뷰를 통해서 일관되게 데이터 처리를 할 수가 있다.

그림 2 에서는 BOOK 테이블에 대해 칼럼, 데이터형태, 인덱스, 키(Key)정보에 대한 구조도를 나타내고 있다. XML 뷰를 제공하기 위해 Oracle XSU(XML SQL Utility) for Java 를 사용했다. XSU 는 다음과 같은 자바 클래스를 제공한다. OracleXMLQuery 클래스는 SQL 질의의 결과 또는 JDBC ResultSet 객체 또는 XMLDataset 객체로 부터 XML 문서를 생성해 준다[7]. OracleXMLSave 클래스는 XML 문서를 데이터베이스 테이블에 저장할 수 있는 메소드를 제공해 준다.

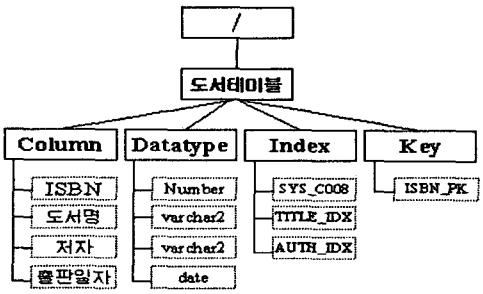

그림 2. 데이터베이스 스키마

그림 3 은 데이터베이스의 스키마를 XSU 에서 제공하는 자바 클래스를 사용하여 XML 문서 형태로 결과를 얻은 것을 Internet Explorer 5.5 에서 출력하였다.

```
<?xml version="1.0" encoding="euc-kr" ?>
- <BOOK_TABLE>
  - <ROW ATTRIBUTE_ID="1">
      <칼럼>ISBN</칼럼>
      <데이타형태>NUMBER</데이타형태>
      <길이>22</길이>
      <순서>1</순서>
      <인덱스>SYS_C00887</인덱스>
      <키>Primary Key</키>
    </ROW>
  - <ROW ATTRIBUTE_ID="2">
      <칼럼>TITLE</칼럼>
      <데이타형태>VARCHAR2</데이타형태>
      <길이>20</길이>
      <순서>2</순서>
      <인덱스>TITLE_IDX</인덱스>
      <키 />
    </ROW>
  - <ROW ATTRIBUTE_ID="3">
      <칼럼>AUTHOR</칼럼>
      <데이타형태>VARCHAR2</데이타형태>
      <길이>10</길이>
      <순서>3</순서>
      <인덱스>AUTHOR_IDX</인덱스>
      <키 />
    </ROW>
  - <ROW ATTRIBUTE_ID="4">
      <칼럼>PUPDATE</칼럼>
      <데이타형태>DATE</데이타형태>
      <길이>7</길이>
      <순서>4</순서>
      <인덱스 />
      <키 />
    </ROW>
</BOOK_TABLE>
```

그림 3. XML 로 표현된 데이터베이스 스키마

위의 그림 3 과 같이 XML 문서에 데이터베이스 스키마의 정보를 표현할 수 있다면 역으로 XML 문서로부터 데이터베이스 스키마를 생성할 수 있다. XML 문서에서 사용되는 태그의 종류, 순서, 발생횟수등의 정보를 표현하기 위해 사용되는 DTD 는 데이터형태를 지원하지 않기 때문에 W3C 에서 XML 스키마 언어를 제안하였다.

XSU 를 이용함으로써 자바가 수행 가능한 데이터베이스-어플리케이션서버-클라이언트 모든 영역에서 XML 문서를 조작(manipulate)할 수 있다. XML 문서를 데이터베이스 테이블에 저장하는 부분에서 현재 XSU 는 W3C 의 DOM(Document Object Model) 워킹그룹에서 표준안으로 만든 DOM 레벨 1 권고안을 기본적으로 사용하고 있어서 XML 문서의 크기가 클 경우 OracleXMLSave 클래스를 이용하면 XML 문서 트리를 만들고 탐색하기 위해 시스템 자원(메모리, CPU)를 과도하게 사용하는 문제가 있다. 따라서 본 고에서는 XML 문서의 저장 부분은 XML 문서의 크기에 관계없이 효율적인 처리가 가능하도록 하기 위해 이벤트 기반(event-based)의 API 인 SAX(Simple API for XML)와 트리 기반의 API 인 DOM 을 같이 사용하는 아래와 같은 접근방식을 취했다.

(1) XML 문서를 반복되는 부분문서(subdocument)로 나누어서 SAX 이벤트가 각각의 부분문서의 제어가 가능하도록 한다.
(2) 각각의 부분문서에 대해 DOM 문서를 생성해서 생성된 문서를 처리하는 메소드를 호출한다.

SAX, DOM 을 같이 사용함으로써 크기가 큰 XML 문서를 전부 메모리에서 처리할 필요가 없고 각각의 부분문서를 처리하기 위해 DOM API 를 사용할 수가 있다.

5. 결론

지금까지 XML 을 객체-관계형 데이터베이스에서 저장, 조작하는 방법과 응용 프로그램에게 일관된 뷰를 제공하기 위해 XML 뷰를 제시하였다. 개발자는 XML 뷰를 통해 XML 저장시스템의 종류에 관계없이 투명하게 XML 기반 질의를 함으로써 데이터 처리를 할 수가 있다. XML 문서는 전용의 XML 저장소에 저장될 수 있고 XML 문서를 저장하고 조작하기 위해 기존의 객체-관계형 데이터베이스에 XML 을 고려한 기능 확장을 통해서 저장될 수 있다. 현재는 객체-관계형 데이터베이스 시스템을 사용할 경우 XML 문서 또는 데이터를 저장하기 위해 칼럼또는 LOB(Large Object)을 이용하고 데이터베이스 질의 언어인 SQL 을 사용하여 데이터 처리를 하지만 향후에는 저장시스템의 종류에 관계없이 투명한 XML 뷰를 제공하여 XML 기반의 질의가 가능해야 한다.

참고문헌

[1] Ronald Bourret, http://www.rpbourret.com/xml/XMLAndDatabases.htm, 2000.

[2] S. Abiteboul, P. Buneman, and D. Suciu, Data on the Web: From Relations to Semistructured Data and XML, Morgan Kaufmann Publishers, San Francisco, CA, 1999.

[3] P. Florescu and D.Kossman, "Storing and Querying XML Data using an RDBMS," IEEE Data Engineering Bulletin 22(2), pp.27-34, 1999.

[4] J.Shanmugasundaram, K. Tufte, C.Zhang, G.He D. J. DeWitt, and J. F. Naughton, "Relational Databases for Querying XML Documents: Limitations and Opportunities," Proc. Of 25th Int'l Conf. On VLDB, Edinburgh, Scotland, UK, pp.302-314, 1999.

[5] Ronald Bourret, http://www.rpbourret.com/xml/XMLDatabaseProds.htm, 2000.

[6] Steve Muench. "Building Oracle XML Applications", First Edition. O'Reilly, 2000.