

XPATH를 이용하여 관계형 데이터를 XML 문서로 출판

남궁숙⁰ 홍의경
서울시립대학교 컴퓨터통계학과 데이터베이스 연구실
{namgung⁰, ekhong}@venus.uos.ac.kr

Publishing Relational Data to XML Using XPATH

Nam Gung Suk⁰ Eui Kyeoung Hong
Department of Computer Science and Statistics, University of Seoul

요 약

XML은 확장성, 호환성, 정보의 구조화의 우수성으로 인터넷상에서 정보 교환의 표준으로 자리 매김하고 있다. 그러나 대부분의 비즈니스 데이터는 관계형 데이터베이스 시스템에 저장되어 있고 앞으로도 상당한 기간 동안 그러할 것이다. 이에 따라, 관계형 데이터를 XML 문서로 변환, 출판하기 위한 연구가 활발히 진행 중이다.

본 연구에서 관계형 데이터를 XPATH 질의어를 통하여 검색하고 XML 문서로 변환하기에 적합하도록 관계형 데이터베이스의 저장 스키마를 설계하였다. 그리고, XPATH 질의어를 SQL 문장으로 변환하여 관계형 데이터를 검색하였고, 질의 결과를 DOM 형식의 XML 문서로 생성하였다.

1. 서 론

XML은 다양한 정보들을 표준화되고 통합된 환경에서 교환할 수 있는 기반구조를 제공한다. 이로 인해, XML은 자연과학에서부터 금융, 이동통신, 멀티미디어 및 각종 산업에 이르기까지 다양한 분야에서 활용되고 있다.

그러나, 대부분의 비즈니스 데이터는 관계형 데이터베이스 시스템의 신뢰성, 안정성, 다양하고 우수한 성능에 결합하여 저장되어 있다. 따라서 관계형 데이터를 가능한 XML 언어를 써서 쉽고 효율적으로 XML 데이터로 변환할 필요가 있다[3]. 이러한 변환을 일반적으로 XML 출판이라고 한다.

현재 데이터베이스 업체들은 그들의 시스템에 적합한 XML Publishing 솔루션을 제공하기 위한 연구를 활발히 하고 있으나 아직 그에 대한 표준은 제정되어 있지 않고 각각의 솔루션은 상당히 다른 형태로 제공되고 있다.

관계형 데이터를 XML로 바꾸는 것은 두 데이터 모델이 매우 상이하기 때문에 복잡하다. 관계형 데이터는 많은 릴레이션에 여러 제약조건으로 정규화된 형태로 존재하고 관계형 데이터의 스키마는 대부분 공유되지 않는다. 이에 반해, XML 데이터는 중첩적이고 비정규적이며(DTD는 데이터 타입의 개념이 없으며 데이터에 제약조건을 부여할 수 없다. XML 스키마는 몇 가지 데이터 타입을 제공하지만 제약조건은 아직 완벽한 형태가 아니다.) 그에 대한 스키마는 공유된다.

* 본 연구는 첨단정보기술 연구센터를 통하여 과학재단의 지원을 받았다.

관계형 데이터를 XML로 변환하는 일반적인 방법으로 관계형 데이터 위에 XML 뷰 생성을 들 수 있다. XML view는 응용 프로그램에게 기반 관계형 데이터에 대한 뷰를 제공한다. XML 뷰를 통하여, 비즈니스 협력업체들은 현존하는 관계형 데이터가 마치 어떤 표준 XML 형식이 있는 것처럼 접근할 수 있다.

본 논문에서는 관계형 데이터베이스와 XML의 사상 정보를 담은 XML 스키마를 정의하여 XML 뷰를 생성하였으며 XPath 질의어를 통해 관계형 데이터를 접근할 수 있도록 XPath를 XML 뷰를 이용하여 SQL문으로 변환하였다.

본 논문 구성은 다음과 같다. 2절에서는 XML 출판 기법에 대해 알아보고, 3절에서는 본 논문에서 사용하는 XML 뷰를 생성하기 위한 XML 스키마에 대해 설명하였다. 4절에서는 XPath 질의어를 RDBMS에서 수행 가능한 SQL문으로 변환하는 방법을 제시하였다. 마지막으로 5절에서는 결론 및 향후 연구 방향을 기술하였다.

2. 관련 연구

2.1 가상(Virtual) 대 실체화(Materialize)

관계형 데이터 위에 XML 뷰가 생성된 다음 야기되는 문제는 어떻게 이 뷰를 사용할 것인가이다. 한 가지 단순한 방법은 요청이 일어나는 대로 전체 XML 뷰를 실체화하여 결과로 생긴 XML 문서를 반환하는 것이다. 이 출판 기법에 의해 관계형 데이터가 하나의 크고 유일한 XML 문서로 생성된다. 응용 프로그램이 전체 XML 데이터의 일부만을 요청할 경우에도 항상 전체 뷰를 얻게 되므로 소모적이라 할 수 있다[5].

가상 XML 출판 기법은 XML 질의를 하나의 XML 가상 뷰에 적용하여 사용자 응용 프로그램에서 실제로 필

요로 하는 XML 데이터를 가져온다.

실체화 XML 출판 기법은 관계형 엔진의 간섭 없이 모든 데이터를 접근할 수 있으나 주기적으로 XML 뷰를 새로 고쳐야 할 필요가 있다.

가상 XML 출판 기법은 데이터의 신선도를 보장하지만 관계형 엔진의 처리력을 바탕으로 하므로 XML 질의를 SQL문으로 바꾸는 복잡한 변환이 요구된다.

2.2 미들웨어 시스템

관계형 데이터를 XML 데이터로 사상 방법을 결정하기 위해 관계형 데이터베이스와 사용자 응용 프로그램간에 작용하는 미들웨어 시스템이 필요하다. 이 미들웨어 시스템은 XML 뷰를 정의하고 관리하며 XML 질의어를 SQL문으로 변환하여 데이터베이스 시스템에 전달한다. 또한, 데이터베이스 시스템으로부터 반환되는 질의 결과를 XML 형태로 변환한다.

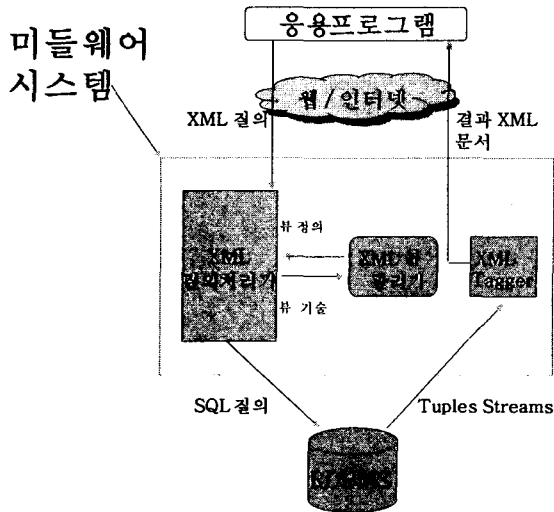


그림 1 미들웨어 시스템의 구조

두 가지 주요 미들웨어 시스템으로 IBM의 XPERANTO와 SilkRoute가 있다[4,5].

XPERANTO는 객체 관계형 데이터베이스 상에서 단일 XML 질의 인터페이스를 제공한다. XML 뷰는 XML 스키마의 타입 시스템을 따르는 XML 질의어로 정의하므로 사용자는 XML과 XML 질의어에만 익숙하면 된다. SQL 테이블과 질의어를 모르고도 데이터베이스 내용을 XML 데이터로 질의하거나 재구성할 수 있다.

SilkRoute는 구조면에서 XPERANTO와 많은 부분이 유사하나 SilkRoute에서 XML 뷰는 RXL(Relational to XML Transformation Language)이라 불리는 선언적인 질의어를 사용하여 정의한다. RXL은 관계형 데이터베이스의 데이터를 추출하여 적절한 엘리먼트와 애틀리뷰트의 태그명을 덧붙인다. 엘리먼트와 애틀리뷰트를 서로 중첩시켜 구조화하고 응용프로그램의 방식에 맞게 값을 처리하여 XML 문서를 만든다.

3. XML 뷰의 생성

본 논문에서는 XML 문서의 구조와 문서내의 데이터에 대한 제약조건들을 기술하는 XML 스키마[1]에 관계형 스키마와의 관계 정보를 추가한 XML 기반 스키마를 정의하였다. 애틀리뷰트와 엘리먼트에 그와 연계되는 관계형 테이블과 열 정보를 추가하여 XML 스키마를 나타냈고 이 스키마를 이용하여 XML 뷰를 생성하는 미들웨어 시스템을 설계하였다.

다음의 예제는 관계형 데이터베이스 스키마를 바탕으로 생성될 XML 문서의 계층관계를 정의한 XML 스키마이다.

```
Orders(order_id(varchar(10)), customer_id(varchar(20)),
item_id(varchar(20)), ...)
Customers(customer_id(varchar(20)), customer_name(varchar(20)),
sex(varchar(2)), address(varchar(50)), ...)
Items(item_id(varchar(20)), item_name(varchar(20)),
category(varchar(20)), ...)
```

그림 2 관계형 스키마

```
<complexType name="customerType">
  <attribute name="customer_id" mtable="customers"
    mcolumn="customer_id"/>
  <element name="sex" mtable="customers" mcolumn="sex"/>
  <element name="address" mtable="customers"
    mcolumn="address"/>
</complexType>
<element name="order" mtable="orders">
  <complexType>
    <attribute name="order_id" mtable="orders"
      mcolumn="order_id"/>
    <element name="customer" type="customerType"
      key_table="orders" key_column="customer_id"
      foreign_table="customers" foreign_column="customer_id" />
    <element name="item_id" mtable="orders" mcolumn="item_id"
      />
  </complexType>
</element>
```

그림 3 XML 스키마

*mtable*은 해당 엘리먼트나 애틀리뷰트가 사상되는 데이터베이스 테이블을, *mcolumn*은 데이터베이스 테이블의 열을 나타내고 *foreign_table*과 *foreign_column*은 참조관계 엘리먼트의 관계형 외래 테이블과 외래키를 나타낸다. 엘리먼트 *order*는 데이터베이스의 *orders* 테이블과 사상되고 부엘리먼트 *customer*는 *customers* 테이블과 사상된다. *customer*는 *order*에 속해 있기 때문에 두 테이블은 참조 관계에 있게 된다. *customers* 테이블의 *customer_id*는 외래키가 되고 *orders* 테이블의 *customer_id*는 기본키가 된다.

위와 같이 스키마를 기술하면 미들웨어 시스템은 이를 바탕으로 관계형 데이터베이스 테이블에 엘리먼트와 애틀리뷰트의 경로 정보와 연관되는 관계형 요소를 기록하여 그림 4와 같은 XML 뷰를 생성한다.

Path 테이블

id	Path	mtable	mcolumn	pid	kid
1	/order	orders			
2	/order/@order id	orders	order id	1	
3	/order/customer	customers		1	1
4	/order/customer/customer id	customers	customer id	3	1
5	/order/customer/sex	customers	sex	3	1
6	/order/customer/address	customers	address	3	1
7	/order/item id	items	item id	1	

Key 테이블

id	key_table	key_column	foreign_table	foreign_column
1	orders	customer_id	customers	customer_id

그림 4 XML 뷰 테이블

이 XML 뷰는 관계형 데이터베이스 테이블에 저장되고 주기억장치에 적재되어 사용한다.

Path 테이블에는 엘리먼트나 애트리뷰트의 경로 표현, 경로 id, 부모 id, 연계 관계의 관계형 테이블과 열명을 저장하고 Key 테이블에는 관계성을 갖는 테이블명과 열명을 저장한다. Path 테이블의 kid는 참조 관계가 있는 엘리먼트의 관련 Key 테이블의 id를 나타낸다.

4. XPath 질의어 처리

XPath는 XSLT 1.0과 함께 W3C에서 XPath 1.0으로 권고된 트리 탐색 언어로서 제대로 갖춰진 질의어는 아니지만, XML 트리 구조 탐색에 대한 기초를 제공한다 [2]. XPath는 URL 경로 표기법을 사용하여 XML 문서의 계층적인 구조를 논리적으로 탐색하는데 위치 경로(location path)를 사용하여 노드의 경로를 나타낸다. 이 위치 경로는 기초부(basis)와 술어부(predicate)로 나누어진다. 기초부는 문맥 노드(context node)와 경로 단계로 선택된 노드간의 관계를 명시하는 축(axis)과 노드 검사(node test)부분으로 구성된다. 술어부는 대괄호([,])로 묶인 표현식을 말하며, 축에 관련된 노드 집합을 새로운 노드 집합으로 만들기 위한 필터링 작업을 한다.

XML 사용자는 XPath 질의어를 통해서 하위 시스템에 상관없이 XML문서에 대한 검색 질의를 하고자 한다. 따라서 미들웨어 시스템은 XPath 질의어를 XML 뷰를 이용하여 SQL 문으로 변환하여 관계형 데이터베이스에 전달한다. 변환 방법은 질의문에서 문맥 노드와 술어부를 추출한다. 이 정보와 XML 뷰를 검색하여 관계형 데이터베이스에 넘겨줄 적합한 SQL 문을 생성한다. 예를 들어 "/order/customer[sex='여']"라는 XPath 질의어를 보자. 위 질의는 customer_id가 'cust001'인 order의 주소를 검색하는 XPath 질의이다.

먼저 질의의 적합성을 검증하고 문맥 노드, "/order/customer"와 술어부, "/order/customer/sex"를 추출한다.

문맥 노드와 일치하는 Path 테이블의 Path를 검색하여 kid값이 있으면 Key 테이블의 해당 id를 검색하여 조인조건을 생성한다. Path의 id를 pid로 갖는 레코드를 검색하여 선택할 열의 집합을 구성하고 자식 id에 kid값

이 존재하면 역시 조인조건을 생성한다. 예제에서 문맥 노드와 일치하는 Path는 id값이 5인 "/order/customer"이다. kid값이 1이므로 Key 테이블의 해당 레코드를 검색하여 orders와 customers의 조인조건을 생성한다. 그리고, 자식노드를 검색하여 customer_id, sex, address를 선택절에 담는다.

술어부와 일치하는 Path를 검색하여 SQL의 조건절을 생성한다. 최종적으로 다음과 같은 질의문이 생성된다.

```

SELECT c1.customer_id, c1.sex, c1.address
FROM orders o1, customers c1
WHERE o1.customer_id = c1.customer_id
and c1.sex = '여'
    
```

위의 SQL 문을 관계형 데이터베이스에서 수행한 결과값을 다시 XML 뷰를 이용하여 XML 문서로 생성한다.

5. 결론 및 향후 연구 방향

본 논문에서는 XML 스키마를 활용하여 관계형 데이터베이스와 XML의 사상 정보를 저장하는 관계형 테이블 형태의 XML 뷰를 생성하였다. 생성된 XML 뷰는 XML의 구조 검색 정보와 관계형 정보를 가진다.

관계형 데이터를 검색하기 위한 질의어로 XML의 XPath를 사용하였고 이는 미들웨어 시스템을 통해 XML 뷰를 이용하여 SQL 문으로 변환하였다.

변환된 SQL 문을 관계형 데이터베이스에 전달하여 질의를 수행하고 결과값을 미들웨어 시스템에 반환하여 XML 뷰를 이용하여 최종적으로 XML 문서로 변환하였다.

향후에는 다른 XML 출판 기법과의 성능 평가를 통해 성능을 개선, 보완하고 XQuery 언어를 지원하여 폭넓고 다양한 질의를 처리할 수 있게 할 예정이다.

6. 참고 문헌

[1] H. S. Thompson, D. Beech, M. Maloney, and N. Mendelsohn, "XML Schema Part I: Structures," World Wide Web Consortium (W3C) Working Draft, <http://www.w3.org/TR/xmlSchema-1>, 2001.

[2] J. Clark and S. DeRose, "XPath 1.0 : XML Path Language," W3C Recommendation, <http://www.w3.org/tr/XPath>, 1999.

[3] J. Shanmugasundaram, et al., "Efficiently Publishing Relational Data as XML Documents," VLDB Conference, Cairo, Egypt, September 2000.

[4] M. Carey, et. al., "XPERANTO: Middleware for Publishing Object-Relational Data as XML Documents," WebDB Workshop, Dallas, May 2000.

[5] Mary F. Fernandez, Atsuyuki Morishima, Dan Suciu, and Wang Chiew Tan, "Publishing Relational Data in XML: the SilkRoute Approach," IEEE Data Engineering Bulletin 24(2), pp.12-19, 2001.