

자료 저장소 래퍼에서 XML 뷰 기반의 XML 스키마 관리 시스템*

정무장^{0*} 박진수^{**} 정채영^{**} 이미영^{***} 강현석^{**} 배종민^{**}
^{**}경상대학교 컴퓨터학과 / 컴퓨터정보통신연구소
^{***}한국전자통신연구원

{pandora, lanez04, wcdi}@rtp.gsnu.ac.kr mylee@etri.re.kr (hskang, jmbae}@nongae.gsnu.ac.kr

An XML Schema Manager based on the XML View in a Storage Wrapper

Mu-Jang Jung^{0*} Jin-soo Park^{**} Chae-yeong Jeong^{**} Mi-young Lee^{***} Hyun-syug Kang^{**} Jong-min Bae^{**}

^{**}Department of Computer Science, Gyeongsang National University /
 Computer Information and Telecommunications Research Institute

^{***}Electronics and Telecommunications Research Institute

요약

미디어터 기반의 데이터베이스 통합 방법론에서 래퍼는 미디어터에게 지역 데이터베이스의 스키마를 전달해야 한다. 여기서는 지역 데이터베이스의 스키마를 XML 스키마로 변환하여 미디어터에게 전달하는 래퍼 시스템에서의 스키마 관리기에 대하여 논한다.

이를 위하여 관계형 모델의 지역 데이터베이스에서, 지역 데이터베이스 스키마를 XML 스키마로 변환할 때 고려할 사항을 논한다. 그리고 XML 질의어인 XQuery로 XML 뷰 정의 언어를 사용할 때, XQuery로 표현된 XML 뷰를 XML Schema로 표현된 XML 스키마로 변환하는 방법을 제시한다.

1. 서론

데이터베이스 통합 방법론 중의 하나인 mediator-wrapper 시스템에서는 지역 데이터베이스에 있는 데이터가 물리적으로 이동되는 것이 아니라, 각 지역 데이터베이스의 내용을 가상적으로(virtual) 통합한(integrated) 뷰를 미디어터(mediator)가 유지한다. 그리고 사용자는 통합된 뷰를 통해서 질의한다. 이를 위해서는 각 지역 데이터베이스와 직접 대화하는 자료 저장소 래퍼(wrapper)가 필요하다. 즉, 래퍼는 각 지역 데이터베이스마다 하나씩 존재하여, 지역 데이터베이스에 대한 내용을 미디어터에게 전달함으로써, 미디어터가 통합된 뷰를 구성할 수 있도록 정보를 제공하고, 미디어터로부터 받은 사용자 질의를 지역 데이터베이스에게 전달하고 그 결과를 미디어터에게 넘겨주는 역할을 한다. 래퍼가 미디어터에게 지역 데이터베이스의 내용을 전달하기 위해서는 각 지역 데이터베이스의 스키마를 표현할 수 있는 하나의 중립적인 모델이 필요한데, XML 기반의 모델이 중요한 대안으로 인정받고 있다.

본 논문에서는 지역 데이터베이스의 스키마 내용을 XML Schema[7]로써 표현한다. XML Schema 명세는 2001년 5월에 발표된 W3C의 권고안에 따른다. 본 논문에서는 먼저, 지역 데이터베이스의 스키마를 XML Schema로 표현하는 기본적인 방법과 고려사항에 대하여 논한다.

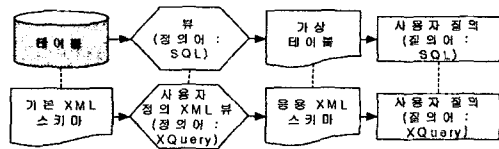


그림 1 데이터베이스 스키마와 XML 스키마의 관계

변환된 XML 스키마(XML schema)는 그 내용이 방대할 수 있기 때문에 사용자가 스키마를 보는 관점을 다시 정의하는 기능이 필요하다. 즉, 사용자가 XML 뷰(XML view)를 정의하는 기능이 필요하다. 이는

SQL의 경우 뷰 테이블을 정의하는 개념과 유사한데, 그 개념은 그림 1과 같다.

사용자 정의 XML 뷰 정의 언어로서 XML 질의어인 XQuery[8]를 사용한다. 그런데 미디어터 입장에서는 XML Schema로 표현된 뷰를 필요로 하기 때문에 XQuery로 표현된 XML 뷰를 XML Schema로 표현하여 미디어터에게 전달해야 한다. 본 논문에서는 XQuery로 표현된 XML 뷰를 XML Schema로 변환하는 방법을 제시한다.

본 논문은 다음과 같이 구성되어 있다. 2장에서 관련 연구에 대해 살펴보고, 3장에서 XML 스키마 관리기의 개략적인 구조를 제시한다. 4장에서 기본 XML 스키마에 대해 살펴보고, 5장에서 응용 XML 스키마에 대해 살펴본다. 6장에서 응용 XML 스키마의 자동 생성에 대해 살펴보고, 마지막으로 7장에서 결론과 향후 과제를 논한다.

2. 관련 연구

관계형 데이터베이스에서 XML 뷰 기반의 질의 처리에 대한 방법이 최근에 연구되기 시작하였다[1, 2]. 그중 XPERANTO[3]에서는 관계형 데이터베이스 스키마를 XML Schema로 표현하고 이를 다폴트 XML 뷰라고 정의한다. 그리고 이를 바탕으로 XQuery를 이용하여 사용자 정의 XML 뷰를 정의하고, 사용자 질의를 이 뷰상에서 처리한다.

XML 문서를 관계형 데이터베이스를 이용하여 사상하는 방법에 대한 연구[5, 6]는 많이 되어 왔는데, XML 문서가 사상되는 규칙에 따라서 XML 뷰를 자동으로 생성하는 방법[4]도 있다.

본 논문에서는 [2, 3, 4]에서 제시된 XML 뷰의 개념을 이용하여 자료저장소 래퍼의 스키마 관리기를 개발함에 있어서 스키마 관리기가 가져야 할 구조를 제시한다. 그리고 미디어터가 관심이 있는 내용, 즉 미디어터 입장에서의 지역 데이터베이스의 대한 뷰를 정의할 수 있도록 XML 뷰를 정의하도록 하는데, 이 때 래퍼가 미디어터에게 전송할 내용은 뷰가 아닌 XML 스키마이기 때문에, 미디어터가 정의한 XML 뷰를 XML 스키마로 자동으로 변환하는 방법을 제시한다.

3. XML 스키마 관리기 기본 구조

스키마 관리기는 미디어터에서 래퍼를 통해 통합된 뷰를 정의하고 질의과정을 수행하기 위해 필요한 XML 스키마를 생성, 수정, 삭제, 등

*본 연구는 한국전자통신연구원의 위탁연구과제인 자료 저장소 Wrapper 개발의 일부로 수행된 결과임

목, 관리하는 기능을 가진다. 스키마 관리기는 자동으로 기본 XML 스키마를 생성하며 이를 바탕으로 사용자가 정의한 사용자 정의 XML 뷰로부터 응용 XML 스키마를 자동 생성한다. 스키마 관리기는 생성된 응용 XML 스키마들을 관리하고 필요에 따라 수정하거나 삭제할 수 있는 기능을 제공한다. 그림 2는 본 논문에서 제안하는 스키마 관리기에서 XML 스키마를 생성하는 개략적인 모습이다.

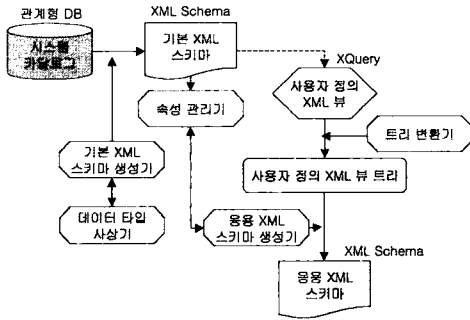


그림 2 스키마 관리기의 XML 스키마 생성 시스템 구조

데이터베이스 스키마의 구조적 의미와 가장 유사한 형태의 구조로 표현한 XML 스키마를 기본 XML 스키마(Base XML schema)라 부른다. 그러나 사용자는 기본 XML 스키마 이외에도 다른 방법으로 사상된 XML 스키마를 원할 수도 있다. 즉, 기본 XML 스키마로 표현된 많은 정보를 이용하기보다는 자신이 필요한 정보만 추출하여 이를 XML 스키마로 표현하기를 원할 수 있다. 이 XML 스키마를 응용 XML 스키마(Application XML schema)라 부른다.

사용자는 이미 생성되어진 기본 XML 스키마를 바탕으로 기본 XML 스키마와 응용 XML 스키마상의 관계를 표현하기 위해 XML 뷰를 정의하게 되는데 뷰 정의 언어로서 XQuery를 이용하여 XML 뷰를 정의하며 이를 사용자 정의 XML 뷰라고 한다. 스키마 관리기는 사용자가 XQuery로 정의한 XML 뷰에 대응하는 응용 XML 스키마를 자동으로 생성한다.

4. 기본 XML 스키마

기본 XML 스키마는 웹퍼가 최하위 수준에서 보는 정보원의 스키마라고 할 수 있다. 관계형 데이터베이스에서 기본 XML 스키마는 데이터베이스의 스키마 구조와 거의 유사한 문서 구조를 가지는 XML 스키마로서 데이터베이스의 시스템 카탈로그를 이용하여 자동적으로 생성된다. 그림 3은 관계형 릴레이션으로부터 자동 생성된 기본 XML 스키마의 일부이다.

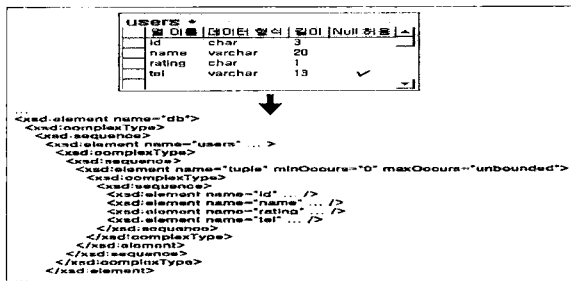


그림 3 관계형 릴레이션으로부터 생성된 기본 XML 스키마

릴레이션의 각 속성들을 XML 스키마로 변환시 속성 엘리먼트들은 데이터 타입을 가지게 된다. 각 속성 엘리먼트의 데이터 타입은 XML Schema에서 제공하는 데이터 타입으로 변환된다. 이때 데이터베이스

와 XML Schema에서 제공하는 데이터 타입 사이에 차이가 존재함으로써 이들간의 적절한 변환이 필요하다. XML Schema에서는 기본 데이터 타입 이외에 데이터 형의 확장이나 제한 또는 변환을 통해 두 데이터 타입 사이의 사상을 가능케 한다. 표 1은 MS-SQL SERVER 2000에서 제공하는 데이터 타입과 XML Schema상에서 제공하는 데이터 타입간의 사상을 나타내는 내용의 일부이다.

MS-SQL 2000	XML Schema
bit	<xsd:element> <xsd:simpleType> <xsd:restriction base="xsd:unsignedByte"> <xsd:minInclusive value="0"/> <xsd:maxInclusive value="1"/> </xsd:restriction> </xsd:simpleType> </xsd:element>
decimal	<xsd:element type="xsd:decimal"/>
float	<xsd:element type="xsd:double"/>
int	<xsd:element type="xsd:int"/>
smallint	<xsd:element type="xsd:short"/>

표 1 데이터베이스와 XML Schema상의 데이터 타입 사상

데이터베이스의 데이터 타입과 XML Schema 데이터 타입의 차이로 인해 이 둘 사이의 정확한 변환은 어렵다. 또한 데이터베이스에서 지원하는 바이너리 타입 형태의 데이터 타입은 XML Schema상에서 표현하기가 어렵고 또한 표현한다 하더라도 후에 데이터베이스의 내용을 XML 문서로 변환시 실제 데이터의 내용을 보여주기가 어렵다. 따라서 바이너리 타입의 데이터 정보는 기본 XML 스키마에 표현되지 않도록 한다. 또한 데이터 타입 중 사용자 정의 타입의 경우에는 현재 지원하지 않고 XML Schema의 anyType으로 변경한다.

경로식	XML 뷰상의 경로식
NULL 값 허용 여부	NULL 값 허용 여부
기본 데이터 타입	속성 엘리먼트의 기본 데이터 타입
부가 데이터 Facets	데이터 타입의 확장 또는 제한

표 2 속성 관리기가 관리하는 자료구조

이 때 속성 엘리먼트에 해당하는 내용은 나중에 새로운 응용 XML 스키마 생성시에 데이터 타입을 얻는데 이용할 수 있도록 속성 관리기에서 관리한다. 속성 관리기에서 관리하는 자료 구조는 표 2와 같다.

5. 응용 XML 스키마

응용 XML 스키마는 기본 XML 스키마를 바탕으로 사용자가 정의한 가상 문서에 대한 스키마이다. 즉, 기본 XML 스키마가 데이터베이스의 스키마 내용을 표현한 것이라면 응용 XML 스키마는 현재 존재하는 데이터베이스에서 사용자가 임의로 생성 가능한 XML 문서의 스키마를 표현한 것이다. 그림 4는 사용자 정의 XML 뷰의 예이다.

```
<Auction>
  { FOR $x IN view("db")/users/tuple
  RETURN
  <Users>
    <Name ID -{<$/id/text()>} $x/name/text() </Name>
    { FOR $y IN view("db")/bids/tuple
    WHERE $y/userid = $x/id
    RETURN
    <Bids>
      <ItemNumber> $y/itemno/text() </ItemNumber>
      <Bid> $y/bid/text() </Bid>
    </Bids> }
    <Rating> $x/rating/text() </Rating>
  </Users> }
</Auction>
```

그림 4 사용자 정의 XML 뷰 예

응용 XML 스키마는 기본 XML 스키마와 마찬가지로 XML Schema

를 이용하여 표현된다. 기본 XML 스키마가 데이터베이스의 시스템 카탈로그를 이용하여 자동으로 생성되는 반면에 응용 XML 스키마의 생성을 위해서는 사용자가 XQuery를 이용하여 XML 뷰를 정의하게 된다. 사용자 정의 XML 뷰는 기본 XML 스키마와 응용 XML 스키마 사이의 사상 관계를 표현한다.

XML 뷰를 정의하기 위한 XQuery는 XML 뷰의 특성상 XQuery 표현에 몇가지 제한이 필요하다. 첫번째는 경로식에서 자손(Descendant) 표기식에 대한 제한이고 두번째는 경로식에서 숨은 경로 표기식에 대한 제한이다. 마지막은 SORTBY() 함수 사용에 대한 제한이다.

경로식에서의 자손 표기식이란 XQuery문의 표현식중 "/"를 의미하고, 숨은 경로 표기식이란 XQuery문의 경로식에서 항상 하위에 자식 엘리먼트를 내포할 수 있는 가능성이 있는 것을 의미한다. 이에 대한 제한은 XML 뷰의 모든 경로식은 명시적으로 정확하게 표현되어야 하기 때문이다. 즉, 사용자의 관점이 개입된 XML 뷰에는 기본 XML 스키마상의 정확한 경로가 표현되어야 한다. 또한 경로식을 표현할 때 text() 함수를 이용하여 원하는 값만을 가져오도록 명시적으로 표현해야 한다. 이는 이후에 사용자의 질의에 의해 반환되는 결과값을 명시적으로 정확히 표현하기 위함이다.

또한 XML 뷰는 실제 데이터를 가지는 것이 아니라 가상적인 문서에 대한 구조를 의미하므로 그 문서 내용상의 정렬에 해당하는 SORTBY는 아무런 의미를 가지지 못한다.

6. 응용 XML 스키마의 자동 생성

사용자 정의 XML 뷰는 트리 변환기에 의해 트리 형태로 변환된다. 뷰 트리를 생성하기 위해서 먼저 가상 루트를 만든다. 사용자 정의 XML 뷰에 나타나는 엘리먼트를 하나의 노드로 만들어 가상 루트 아래에 추가한다. 또한 FLWR 절이 나타날 때마다 반복 노드를 만들어 추가한다. 각 노드는 노드 타입에 따라 구분된다. 트리의 각 노드는 표 3과 같은 자료 구조를 가진다.

속성	내용
이름	엘리먼트 이름
타입	노드 타입
부모노드	현재 노드의 부모노드 포인터
첫번째 자식노드	현재 노드의 첫 번째 자식노드 포인터
다음형제 노드	현재 노드의 다음형제 노드 포인터
값	XML 뷰상의 경로식
에트리뷰트	에트리뷰트 이름, 에트리뷰트 값

표 3 사용자 정의 XML 뷰 트리 노드 구조

'이름' 필드는 사용자 정의 XML 뷰에 나타나는 엘리먼트의 이름을 의미한다. '부모 노드' 필드, '첫 번째 자식 노드' 필드 그리고 '다음 형제 노드' 필드는 현재 엘리먼트를 기준으로 부모 엘리먼트, 첫 번째 자식 엘리먼트 그리고 다음 형제 엘리먼트를 의미하며 각각 해당 노드의 포인터 값을 가진다. '타입' 필드는 트리를 구성하는 루트 노드인지 반복 노드인지 엘리먼트 노드인지로 분류한다. '값' 필드는 엘리먼트 노드이면서 단말 노드일 경우 가지게 되며 노드의 경로식이나 상수를 가지게 된다. 마지막으로 '에트리뷰트' 필드는 엘리먼트에 에트리뷰트가 존재할 경우 값을 가지며 여기서 값은 에트리뷰트 이름과 에트리뷰트 값을 의미하며 리스트로 구성된다.

그림 5는 그림 4의 사용자 정의 XML 뷰를 트리로 변환한 예이다. 사용자 정의 XML 뷰 트리는 문서의 구조적 정보를 표현하고 있다. 사용자 정의 XML 뷰에 나타나는 조건절은 뷰 트리상에 아무런 영향을 미치지 못한다. 이는 응용 XML 스키마는 실제 XML 문서의 내용이 아니라 XML 문서의 DTD 정보이므로 의미상의 조건은 문서의 구조에는 영향을 미치지 못하기 때문이다.

또한 이 구조적인 정보에 부가적으로 XML 스키마의 생성을 위한 엘리먼트들의 데이터 타입이 필요하며, 이를 얻기 위해 속성 관리기를 이용하게 된다. 변환된 뷰 트리를 순회하다 만나게 되는 단말 노드나 노드가 가지는 에트리뷰트의 경우에는 데이터 타입이 필요하게 되는데,

이 때 속성 관리기로부터 일치하는 데이터 타입을 얻게 된다.

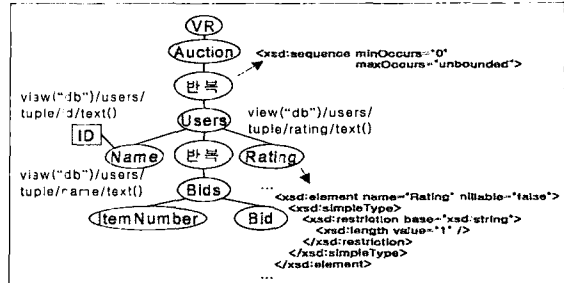


그림 5 사용자 정의 XML 뷰 트리 예

사용자 정의 XML 뷰는 기본 XML 스키마를 기반으로 하기 때문에 사용자 정의 XML 뷰에 나타나는 데이터 타입 정보들은 모두 기본 XML 스키마에서 얻을 수 있다. 변환된 뷰 트리상의 노드가 가지는 경로식과 속성 관리기에서 관리하는 기본 XML 스키마상에 나타나는 경로식을 비교하여 일치된 엘리먼트 데이터 타입을 결정한다. XQuery에서 지원되는 집합 함수에 따라 응용 XML 스키마상에 나타나는 데이터 타입은 변경될 수 있으며 사용자가 새롭게 정의하는 상수값들은 데이터 타입에 상관없이 모두 String 형으로 변환한다.

7. 결론 및 향후 과제

본 논문에서는 XML을 기반으로 이질의 분산 데이터베이스를 통합할 수 있는 랩퍼를 위한 XML 스키마 관리기를 제안하였다. 스키마 관리기는 관계형 데이터베이스 시스템에서 기본 XML 스키마를 바탕으로 사용자가 정의한 사용자 정의 XML 뷰를 이용해 사용자가 원하는 응용 XML 스키마를 생성한다.

스키마 관리기는 실제 데이터베이스에 존재하는 방대한 자료를 사용자가 원하는 관점에서 볼 수 있도록 XML 뷰를 관리하는 역할을 해주며, XML Schema로 정의된 응용 XML 스키마는 사용자에게 XML 뷰의 의미상의 구조를 숨겨 질의를 하는 사용자에게 실제 데이터베이스의 스키마의 구조를 숨겨준다. 또한 기본 XML 스키마를 응용 XML 스키마와 동일한 XML Schema로 표현함으로써 필요에 따라 사용자가 응용 XML 스키마의 정의없이 질의를 할 수 있는 방법도 가능케 한다.

앞으로 실제 같은 관계형 데이터베이스라 하더라도 시스템 카탈로그의 내부 표현이 모두 다르고 지원하는 데이터 타입도 다르기 때문에 이를 해결하기 위한 일반적인 방법에 대한 연구가 필요하다.

참고문헌

- [1] M. Fernandez, W. Tan and D. Suciu, "SilkRoute: Trading between Relations and XML", WWW9, pp. 723-745, 2000.
- [2] J. Shanmugasundaram, J. Kiernan, E. Shekita, C. Fan and J. Funckeburk, "Querying XML Views of Relational Data", VLDB Conference, pp. 261-270, 2001.
- [3] M. Carey, D. Florescu, Z. Ives, Y. Lu, J. Shanmugasundaram, E. Shekita and S. Subramanian, "XPERANTO: Publishing Object-Relational Data as XML", Workshop on the Web and Databases (WebDB), pp. 105-110, May 2000.
- [4] J. Shanmugasundaram, E. Shekita, J. Kiernan, R. Krishnamurthy, E. Vigles, J. Naughton and I. Tatarinov, "A General Technique for Querying XML Documents using a Relational Database System", SIGMOD Record 30(3), pp. 20-26, September 2001.
- [5] A. Deutsch, M. Fernandez and D. Suciu, "Storing Semi-structured Data with STORED", SIGMOD Conference, pp. 431-442, May 1999.
- [6] D. Florescu and D. Kossman, "Storing and Querying XML Data using an RDBMS", IEEE Data Engineering Bulletin, 22(3), pp. 27-34, 1999.
- [7] World-Wide Web Consortium, "XML Schema Part 0: Primer", <http://www.w3.org/TR/xmlschema-0/>.
- [8] World-Wide Web Consortium, "XQuery 1.0: An XML Query Language", <http://www.w3.org/TR/xquery/>.