

XML 뷰를 지원하는 래퍼에서 트리 기반의 XML 뷰 합성 모델

강동완*, 배종민*

*경상대학교 컴퓨터과학과

e-mail:wanni76@naver.com, jmbae@nongae.gsnu.ac.kr

A Tree-Based View Composition Model in Storage Wrapper Supporting XML Views

Dong-Wan Kang*, Jong-Min Bae*

*Dept of Computer Science, GyeongSang National University

요 약

XML 뷰 기반의 래퍼 시스템은 지역 데이터베이스에 대한 XML 뷰 정의를 지원하고, 사용자는 XML 질의어를 이용하여 XML 뷰에 대한 질의를 할 수 있다. 이 때, XML 뷰와 질의간의 합성이 필수적인데, 본 논문에서는 XML 뷰와 사용자 질의를 높은 수준에서 추상화하여 모두 트리로 표현하고, 이것을 이용하여 질의에 나타난 뷰의 경로식에 대한 항해를 통해 합성을 수행하는 모델을 제시한다. 뷰 트리는 가상의 XML 문서의 구조를 그대로 반영하기 때문에 경로식의 항해를 쉽게 하며, XML 뷰의 스키마를 생성하고, 질의 결과 문서를 구성하는 템플릿으로 활용된다. 또한, XML 뷰와 질의의 추상화된 개념적 동일성은 단단계 XML 뷰 정의와 합성을 지원하고, 구현을 용이하게 한다.

1. 서론

XML 기반의 래퍼 시스템은 사용자가 자료 저장소 래퍼(이하 래퍼라 칭함)를 XML 데이터베이스 시스템처럼 사용할 수 있도록 한다. 래퍼는 데이터베이스의 스키마를 XML 스키마로 표현하여 사용자에게 제공하고, 사용자는 래퍼에 대해 XML 질의어를 이용하여 질의할 수 있으며, 질의 결과로 XML 문서를 얻게 된다. 이를 위해서, 래퍼는 데이터베이스에 대한 XML 뷰를 사용자에게 제공한다. XML 뷰는 데이터베이스에 저장된 데이터를 임의의 구조로 맵핑한 가상적인 XML 문서이다. 래퍼는 XML 뷰에 대한 질의를 데이터베이스의 질의로 변환하여 수행하고, 그 결과를 XML 문서로 변환하여 사용자에게 전달한다.

래퍼에서 XML 뷰에 대한 질의 처리를 위해서는 XML 뷰와 사용자 질의간의 합성이 필요한데, 본 논문은 XML 뷰와 사용자 질의를 개념적으로 추상화하여 모두 트리로 표현하고, 이것을 이용하여 질의에 나타난 뷰의 경로식에 대한 항해를 통해 합성

을 수행하는 모델을 제시한다. 제시된 트리 모델은 임의의 XQuery[1] 표현식으로부터 구성되며, 가상의 XML 문서의 구조를 그대로 반영한다. 이것은 뷰에 대한 경로식의 항해를 쉽게 하여 합성을 편하게 만들고, 트리를 질의 결과로 생성되는 XML 문서를 구성하기 위한 템플릿으로 활용할 수 있게 하며, XML 뷰의 스키마 생성을 쉽게 한다. 개념적으로 추상화된 XML 뷰와 질의는 동일한 내부 모델로 표현되기 때문에 구현의 중복이 제거되며, 여러 단계에서 정의되는 XML 뷰의 지원과 합성에 대한 구현이 쉬워진다. 본 논문에서는 제시된 합성 모델을 관계형 데이터베이스에 대한 래퍼에서 구현한다. 그러나, 제시된 합성 모델은 XML 뷰에 대해 개념적으로 추상화되어 있어 데이터베이스 모델에 독립적이다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구에 대해 살펴보고, 3장과 4장에서는 각각 XML 뷰와 뷰 트리 모델에 대해 설명한다. 5장에서는 뷰 트리를 이용한 합성 과정을 설명하며, 마지막으로 6장에서는 결론을 제시한다.

2. 관련 연구

관계형 데이터를 XML 문서로 출판하는 미들웨어 시스템으로 SilkRoute[2], XPERANTO[3] 등이 있다.

SilkRoute는 자체적으로 정의한 선언적인 질의어인 RXL(Relational to XML)을 사용하여 XML 뷰를 기술하고, XML-QL로 정의된 사용자 질의어와의 합성을 통해서 새로운 RXL을 생성한다. SilkRoute에서 합성은 템플릿과 스킴 합수로 구성된 뷰 트리를 이용한 패턴 매칭과 질의 재작성을 통해 이루어진다.

XPERANTO는 뷰와 질의를 XQuery를 사용하여 표현하며, 관계형 모델을 확장한 내부 질의 모델인 XQGM(XML Query Graph Model)을 이용하여 합성한다. XPERANTO는 합성 과정에서 질의의 프레디케이트를 뷰로 내려보낸 후, 질의에 나타난 뷰에 대한 항해를 제거하고, 질의에 포함되지 않은 부분들 뷰에서 제거한다.

본 논문은 XQuery를 이용하여 뷰와 질의를 정의하며, XQuery를 질의 결과 XML 문서의 구조를 그대로 반영하는 트리 구조로 구성하여 경로식에 대한 항해를 통해 합성을 수행한다.

3. XML 뷰

본 장에서는 랩퍼에서 XML 뷰에 대한 질의 과정을 예를 들어 설명한다.

그림 1은 온라인 경매 시스템을 위한 데이터베이스의 예이다. users 테이블은 경매에 참가하는 사람들에 대한 정보를 유지하고, items 테이블에는 경매에 나온 물건들에 대한 정보가 저장되며, bids 테이블은 입찰 데이터를 담고 있다. bids 테이블의 u_id와 i_id 필드는 각각 users 테이블과 items 테이블의 id 필드와 연결된다.

users			items				bids		
id	name	rating	id	description	offeror	price	u_id	i_id	bid
101	Kim	A	1001	Bicycle	101	40	101	1002	400
102	Kim	B	1002	Motocycle	102	500	102	1001	35
			1003	Helicopter	102	50000	101	1003	45000

그림 1 온라인 경매 데이터베이스

랩퍼는 지역 데이터베이스에 대한 하위 수준의 XML 뷰를 자동적으로 생성하는데, 이것을 기본 XML 뷰라고 한다. 기본 XML 뷰는 데이터베이스에 저장되어 있는 전체 데이터들을 XML 문서로 맵핑한 것이다. 관계형 데이터를 XML 문서로 맵핑하는 방법에는 여러 가지가 있을 수 있는데, 본 논문에서

는 그림 2와 같이 테이블을 하나의 XML 문서로 대응시키고, 테이블의 스키마 구조를 그대로 XML 문서의 내포 구조로 맵핑한다. 결국, 전체 데이터베이스는 다수의 XML 문서로 맵핑된다.

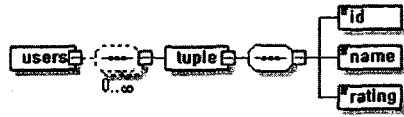


그림 2 테이블을 XML로 맵핑

사용자는 관계형 데이터를 원하는 구조의 XML 문서로 구성하기 위하여 기본 XML 뷰에 대해 사용자 정의 XML 뷰를 정의할 수 있는데, 이것을 응용 XML 뷰라고 한다. 예를 들어, 경매에 참여한 사람들의 이름과 입찰 리스트, 그리고 그들의 신뢰도를 나타내는 XML 문서를 구성하기 위하여 그림 3과 같은 응용 XML 뷰를 XQuery를 사용하여 정의할 수 있다. 사용자는 기본 XML 뷰에 대하여 응용 XML 뷰는 더 높은 추상화를 위하여 다른 응용 XML 뷰에 대하여서도 정의될 수 있다.

```

<Auction> {
  FOR $user IN document("users")/tuple
  RETURN
    <User ID={ $user/id/text()} >
    <Name> $user/name/text() </Name>
    <Bids> {
      FOR $item IN document("items")/tuple,
        $bid IN document("bids")/tuple
      WHERE $bid/u_id = $user/id AND $bid/i_id = $item/id
      RETURN
        <Item>
          <Description> $item/description/text() </Description>
          <Price> $item/price/text() </Price>
          <Bid> $bid/bid/text() </Bid>
        </Item>
      </Bids>
    <Rating> $user/rating/text() </Rating>
  </User>
}
</Auction>
    
```

그림 3 응용 XML 뷰

XML 뷰가 정의되어지면 사용자는 XML 뷰에 대한 질의를 할 수 있다. 그림 4은 그림 3에 있는 응용 XML 뷰가 Auction이라는 이름으로 정의되었을 때, 신뢰도가 "A"인 사람들의 이름과 입찰 리스트를

반환하는 질의이다. 응용 XML 뷰에 대한 질의는 합성을 통하여 기본 XML 뷰에 대한 질의로 형식화되며, 합성된 질의는 기본 XML 뷰와 데이터베이스 간의 맵핑 관계를 기반으로 데이터베이스의 질의로 변환되어 수행된다.

```

FOR $user IN document("Auction")/User
WHERE $user/Rating = "A"
RETURN
  <Result>
    { $user/Name }
    { $user/Bids }
  </Result>
    
```

그림 4 사용자 질의

4. XML 뷰 트리

XQuery로 작성된 응용 XML 뷰와 사용자 질의는 내부적으로 트리로 구성된다. XQuery의 엘리먼트 생성자는 결과 XML 문서의 구조를 결정하는데, 이것을 이용하여 뷰와 질의를 트리로 구성한다. 그림 5는 그림 3의 응용 XML 뷰를 트리로 구성한 것이다.

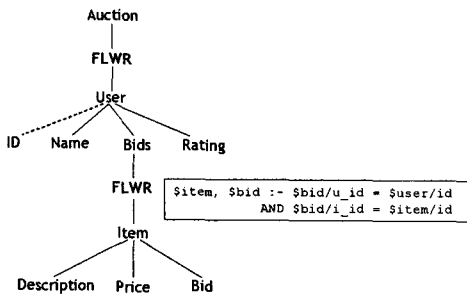


그림 5 응용 XML 뷰 트리

뷰 트리는 XQuery의 엘리먼트 생성자로부터 구성되는 결과 XML 문서의 엘리먼트의 정보를 저장하는 노드와 FLWR 표현식의 질의 정보를 저장하는 노드로 구성된다. 엘리먼트 노드는 엘리먼트의 이름과 데이터 타입, 엘리먼트의 텍스트 데이터와 연결되는 경로식등이 저장된다. FLWR 노드에는 FLWR 질의 변수와 프레디카드, 정렬 조건에 대한 정보가 저장된다. FLWR 노드 하위에는 FLWR 절에 의해서 구성되는 엘리먼트 노드가 추가된다.

기본 XML 뷰는 지역 데이터베이스에 대한 맵핑이므로 응용 XML 뷰와 사용자 질의와는 다르게 XQuery로 정의되어지지 않으며 자동으로 구성되어

진다. 그러나, XML 뷰에 대한 개념적인 추상화를 통하여 모든 XML 뷰를 동일한 데이터 모델로 표현할 수 있다. 데이터베이스에 저장된 데이터를 개념적으로 XML 문서라고 가정하고, 이 XML 문서 전체를 그대로 반환하는 질의를 XQuery로 정의하면, 기본 XML 뷰를 XQuery로 표현할 수 있다.

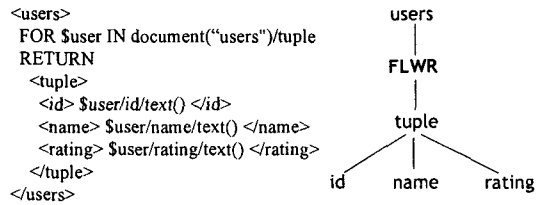


그림 6 users 테이블의 기본 XML 뷰와 트리

그림 6은 그림 1의 경매 데이터베이스의 users 테이블에 대한 XQuery로 작성된 기본 XML 뷰와 뷰 트리를 보여준다. 그림 6에서 보여지는 XQuery는 실제로 작성되어지지 않고 뷰 트리만 자동적으로 구성된다. 이를 위해, 데이터베이스의 시스템 카탈로그가 이용되며, 질의 변환에 필요한 시스템 카탈로그의 정보는 뷰 트리의 노드에 유지된다.

5. 뷰 트리를 이용한 합성

합성은 응용 XML 뷰를 구체화 하지 않고 질의 변환에 필요한 정보를 획득하기 위해 응용 XML 뷰에 대한 질의를 기본 XML 뷰에 대한 질의로 재형식화 하는 과정이다. 본 논문에서 합성은 뷰 트리를 매개로 하여 이루어지는데, 합성 결과로서 뷰 트리들의 노드들 사이에 포인터 연결이 생기며, 질의와 뷰의 프레디카드들이 합쳐진다. 따라서, 사용자 질의를 구성하는 트리의 노드들과 기본 XML 뷰 트리의 노드들은 포인터로 연결되어지며, 이것을 통해 사용자 질의는 기본 XML 뷰에 대한 질의로 표현된다.

본 논문에서는 XQuery를 트리로 구성하는 과정에서 동시에 합성을 수행한다. 뷰 트리에 어떤 노드가 추가될 때, 이 노드와 연관된 경로식에 대한 항해가 이루어지고, 항해 결과로 바인딩된 노드에 대한 포인터 연결을 유지하게 된다. 그림 7은 그림 3의 응용 XML 뷰와 그림 4의 사용자 질의가 뷰 트리표현되어 합성된 모습을 보여준다. 그림 7에서 보는 것처럼 뷰 트리 사이에 노드들간의 포인터 연결이 생성되고, 이러한 포인터 연결을 통해 사용자 질의는 기본 XML 뷰에 대한 질의로 표현될 수 있다.

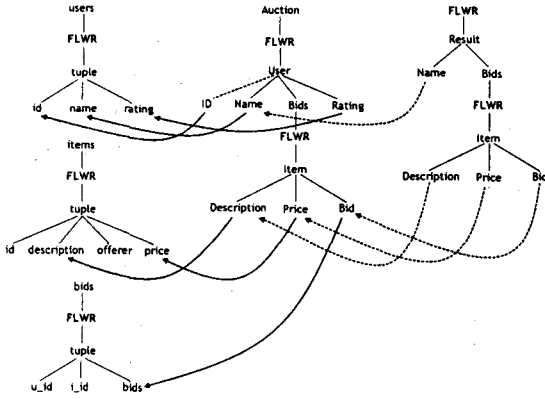


그림 7 XML 뷰 간의 합성

뷰 트리는 가상적인 XML 문서의 구조를 유지하기 때문에, 뷰를 실제화하지 않더라도 XQuery에 표현된 경로식에 대한 향해를 가능하게 한다. 기본적으로 XQuery에 나타나는 경로식에 대한 향해는 실제화된 XML 문서에 대해 이루어지는 것이지만, 가상적인 XML 문서에 대한 향해이므로 XML 문서의 스키마 수준에서 향해가 이루어지는 것이다. 결국, 뷰 트리에 대한 향해의 결과는 실제화된 XML 문서에 포함된 엘리먼트의 메타데이터가 된다. 또한, 이러한 경로식에 대한 향해는 잘못 작성된 경로식에 대한 검출을 쉽게 한다.

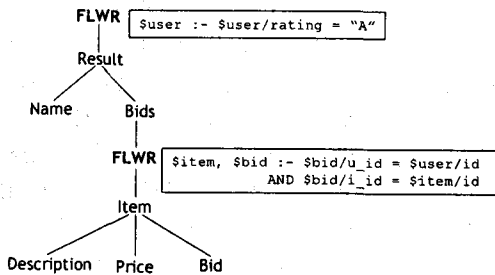


그림 8 합성된 사용자 질의 트리

합성 과정에서 질의와 응용 XML 뷰의 프레디카트는 합쳐진다. 본 논문에서는 뷰 트리에 대한 경로식의 향해 중에 만나는 뷰의 FLWR 노드의 정보는 질의 트리에서 해당 경로식과 연관된 FLWR 노드로 복사되어 재작성된다. 그림 8은 그림 4의 사용자 질의에 대한 트리를 보여주는데, 뷰의 프레디카트가 포함되어 있다. 그리고, XQuery에 나타난 숨은 표기식의 향해 결과로 바인딩된 뷰의 노드가 복사되어

있다.

래퍼는 합성된 질의를 기본 XML 뷰에 저장된 데이터베이스 스키마 정보를 이용하여 데이터베이스의 질의로 변환하여 수행한다.

6. 결론

본 논문에서 제시한 트리 모델은 가상의 XML 문서의 구조를 그대로 반영하므로 결과 XML 문서를 구성하기 위한 템플릿으로 활용할 수 있으며, XML 뷰의 스키마 생성을 쉽게 한다. 또한, 경로식에 대한 쉬운 향해는 합성을 편하게 하며, XML 뷰에 대한 추상화는 다단계 뷰에 대한 정의와 합성을 쉽게 만든다. 즉, 본 논문에서 제시된 트리 모델은 뷰 합성과 스키마 생성, 질의 변환, 결과 문서 생성을 모두 고려한 결과물이다.

또한 본 논문에서 제시한 트리 모델을 이용한 XML 뷰의 합성은 지역 데이터베이스에 독립적이다. 뷰 트리와 이를 이용한 합성 메커니즘은 가상적인 XML 문서에 대한 질의 수준에서 이루어진다. XML 뷰를 지원하는 래퍼는 지역 데이터베이스의 종류에 관계없이 이러한 기능의 구현을 필요로 한다. 그러나, 스키마 생성, 질의 변환, 결과 문서 생성은 지역 데이터베이스에 영향을 받으며 래퍼마다 다른 구현을 필요로 한다. 따라서, 본 논문에서 제시된 트리 모델과 합성 메커니즘은 XML 뷰를 지원하는 래퍼를 구현하기 위한 프레임워크로 활용될 수 있다.

참고문헌

- [1] World-Wide Web Consortium, "XQuery 1.0: An XML Query Language, W3C Working Draft", <http://www.w3.org/TR/xquery/>.
- [2] M. Fernandez, W. Tan, and D. Suciu, "SilkRoute : Trading between Relations and XML", WWW9, pp. 723-745, 2000.
- [3] J. Shanmugasundaram, J. Kiernan, E. Shekita, C. Fan, and J. Funderburk, "Querying XML Views of Relational Data", VLDB Conference, pp. 261-270, 2001.
- [4] 정채영, 최규원, 김영옥, 김영균, 강현석, 배종민, "관계형 데이터베이스에서 XML 뷰 기반의 질의 처리 모델", 정보처리학회논문지 제10-D권 제2호, 2003.