

SQL:1999를 이용한 XPath 질의 처리기의 설계*

황정수⁰ 홍의경

서울시립대학교 컴퓨터통계학과

{jshwang99⁰, ekhong}@venus.uos.ac.kr

Design of XPath Query Processor Using SQL:1999

Jung-Su Hwang⁰ Eui Kyeoung Hong

Department of Computer Science and Statistics, University of Seoul

요 약

인터넷에서 XML은 새로운 정보 환경의 요구사항인 확장성과 문서 관계성의 우수성을 만족하는 표준이다. 이러한 XML 문서 안의 정보 검색을 위해서 XPath 질의어가 널리 사용 중이다. 또한, XML 문서의 체계적인 운용을 위해 데이터베이스에 효율적으로 저장하고 검색하는 연구들이 진행되고 있다.

본 연구에서는 분할 저장 시스템에서 RDBMS 또는 ORDBMS에 XML 문서를 저장하고 검색할 수 있도록 하는 XPath 질의 처리기를 SQL:1999를 이용하여 설계함으로써 좀더 정확하고 빠르게 처리가 가능하도록 하였다.

1. 서 론

최근 웹을 이용한 정보의 교류가 폭발적으로 증가함에 따라 초기 인터넷 웹 페이지를 표현하는 HTML을 가지고는 증가하는 정보의 양을 만족스러운 수준으로 처리하는 것이 어려워졌다. 이에 따라 W3C에서는 기존의 HTML의 단점을 보완하고 SGML의 복잡성을 제거한 XML(eXtensible Markup Language)을 웹 문서의 표준으로 제정하였다. XML은 SGML의 장점인 확장성과 문서 관계성이 뛰어나 새로운 정보 공유 환경을 담당할 새로운 표준으로 자리잡아가고 있다. 이런 XML 문서를 데이터베이스 시스템에 저장하는 기법에 대한 연구가 활발히 진행 중이며 오라클사의 XMLDB 시스템을 비롯한 많은 시스템들이 상용화되어 있다.

또한 다양한 형태로 저장된 XML 문서에서 원하는 데이터를 추출하고 변환하는 작업들이 필요하게 된다. 이를 해결하기 위해서 일반적으로 질의어를 사용하게 된다. W3C에서는 XML 문서에 대한 질의어로서, 구조적 질의어인 XPath를 핵심으로 하는 XQuery를 새로운 질의어의 표준으로 발표하였다[1,3].

따라서 XML 문서를 관계형 데이터베이스 시스템(RDBMS)에 효율적으로 저장하는 시스템을 개발하고, XML사용자가 저장된 XML 문서에서 필요한 정보를 검색할 수 있도록 하기 위해 XQuery 질의어를 지원하는 연구가 필요하다.

본 논문에서는 서울시립대학교에서 설계한 XML 문서 저장 기법과 그것을 기반으로 설계한 XPATH 질의 처리기[6]를 SQL99를 이용하여 좀더 좋은 성능을 가지는 질의 처리기로 재설계하였다.

본 논문은 다음과 같이 구성되었다. 2절에서는 XML 문서 구조 저장 기법, XPath 및 SQL99에 대해 알아보고, 3절에서는 본 논문에서 사용하는 XML 문서 저장 시

스템에 관하여 설명한다. 4절에서는 SQL:1999를 이용한 XPATH 질의 처리기를 설계하였다. 마지막으로 5절에서는 결론 및 향후 연구 방향을 기술한다.

2. 관련 연구

2.1 XML 문서 구조 저장 기법의 분류

데이터 베이스 기술 분야에서는 방대한 양의 데이터를 웹을 통해 공유하기 위한 언어인 XML로 작성된 문서를 효율적으로 DBMS에 저장하고 검색하는 기술에 대한 연구가 활발히 진행되고 있다[3].

그 중 관계형 데이터베이스에 저장하는 방식은 크게 가상 분할 저장 기법과 분할 저장 기법으로 나눌 수 있다. 가상 분할 기법은 XML 문서 전체 내용을 하나의 BLOB (Binary Large Object) 형으로 데이터베이스에 저장하고 XML 문서의 엘리먼트들을 추출하고, 그 엘리먼트가 실제 문서상의 위치 정보를 표현하기 위해서 시작 오프셋과 종료 오프셋을 사용하여 표현한다. 가상 분할 기법은 검색 효율이 우수하지만 문서 일부분의 갱신 작업에 대한 오버헤드가 크기 때문에 일반적으로 검색 위주의 응용 개발 시스템에 주로 사용된다. 반면에, 분할 저장 기법은 XML 문서의 내용을 엘리먼트 단위로 나누어서 저장하고 검색 시 구조 정보를 참조하여 해당 엘리먼트나 하위 엘리먼트들을 재구성하여 검색 결과를 반환하는 기법을 말한다[2].

최근에는 분할 저장기법의 경우 크게 모델 기반 저장 방식과 구조 기반 저장 방식으로 나눌 수 있다. 모델 기반 저장 방식은 모든 XML 문서의 구조를 저장하는 고정된 스키마를 가지고 있어서 특정 XML 문서의 구조에 영향을 받지 않는다. 반면에 구조 기반 저장 방식의 경우는 XSchema나 DTD와 같은 XML 문서의 구조정보를 가지고 테이블을 설계하며, 따라서 특정 XML문서의 구조에 영향을 받는다.

2.2. XPath

XPath는 XSLT 1.0과 함께 1999년 11월 16일 W3C에 XPath 1.0으로 권고(Recommendation)되었다. XPath는 XML 문서의 일부분을 지시하기 위해서 사용된다. XPath는

* 본 연구는 첨단정보기술 연구센터를 통하여 과학재단의 지원을 받았음

URL 경로 표기법을 사용하여 XML 문서의 계층적인 구조를 논리적으로 탐색하며 요소 노드(element node), 속성 노드(attribute node), 값 노드(text node)를 사용하여 트리 구조로 형성된다. 또한 XPath는 각 노드의 문자열 값(string value)을 계산하는 방법을 정의하고 있다[1].

XPath는 노드의 경로를 나타내기 위해 위치 경로(location path)를 사용하는데, 경로 위치 표현방식에는 절대 경로와 상대 경로 두 가지 방식이 있으며 이때 경로 위치를 기초부(basis)와 술어부(predicate)로 나누어 설명할 수 있다.

표 1. XPath의 생략된 표기식

비생략된 문법	생략된 문법
child::	(없음)
attribute::	@
/descendant-or-self::node()	//
parent::node()	..
(문서 루트 노드)	/

본 논문에서는 XPath의 모든 기능을 만족하기보다는 XQuery에서 지원하는 생략된 표기식만을 사용하고 비생략된 표기식과의 비교는 표 1을 통해 확인할 수 있다. 또한, 단일 술어부에 대한 지원을 보장하며, 술어부에 사용되는 비교 연산자로는 관계 연산자만을 사용한다.

2.3 SQL:1999

1986년 IBM System R의 언어였던 SEQUEL을 기반으로 하여 SQL:86이 만들어진 후 관계 데이터베이스의 조인, 스키마 변경 등 다양한 개념을 지원하는 SQL92가 개발되었고 1999년에 객체지향, 객체-관계 데이터베이스의 여러 개념들을 지원하는 언어인 SQL:1999가 제창되었다. SQL:1999는 SQL:92에서 지원하던 관계 데이터베이스의 개념에 객체 개념을 첨부하여 LOB 데이터 유형을 비롯한 여러 복합 데이터 유형들과 새로운 시맨틱들을 지원하는 SQL 표준이다[5].

3. XML 저장 시스템 구조

본 논문에서 native-XML 문서를 DTD에 상관없이 독립적으로 RDBMS에 저장할 수 있도록 저장 스키마를 그림 1과 같이 설계하였다.

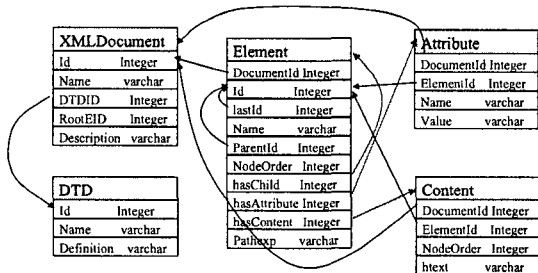


그림 1. XML 저장 스키마

본 시스템에서 XML 문서를 저장하기 위해서 다음과 같은 과정을 거친다. 우선 SAX2 파서를 이용하여 XML 문서를 파싱한다. 그런 후 파싱된 이벤트와 저장 관리기를 이용하여 엘리먼트, 애트리뷰트, 내용 등의 정보를 각각의 테이블에 사상해서 저장한다.

저장 관리기는 SAX 이벤트가 XML 문서에 대해 루트노드부터 DFS 방식으로 발생시킨 노드 이벤트를 가지고 각 엘리먼트에 대해서 ID를 부여한다. 그리고 현 엘리먼트의 경로 위치를 나타내기 위해서 PathExp를 사용한다. 그림 2는 DFS Numbering을 이용하여 ID를 부여한 것과 경로 정보를 추가적으로 표현한 그림이다[6].

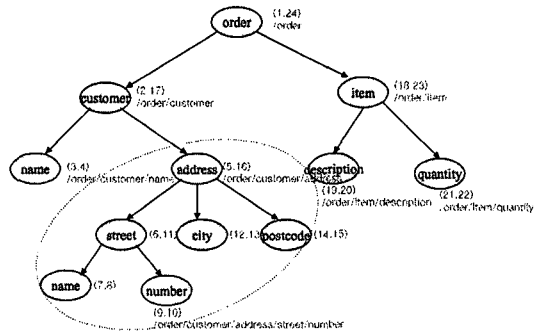


그림 2. DFS Numbering과 PathExp 정보 표현 트리

이렇게 저장된 XML 문서를 XPath를 사용하여 질의할 수 있어야 한다. 다음 절에서는 XML 문서에 대한 XPath질의를 SQL:1999표준 질의로 바꾸어 데이터베이스로부터 XML 데이터를 추출하고 이것을 가지고 XML 문서를 생성하는 방법을 설명한다.

4. 질의 처리기 설계

본 논문에서는 서울시립대학교에서 설계한 XPath 질의 처리기(표현상의 편의를 위해 이후로는 '이전 질의처리기'라 칭한다)를 기반으로 SQL:1999의 새로운 시맨틱을 적용하여 좀더 효율적으로 데이터의 추출이 가능하도록 하는 새로운 XPath 질의 처리기를 설계하였다.

질의 처리기 구조는 그림 3과 같이 처리하기 위해 크게 세 가지 단계로 구성된다.

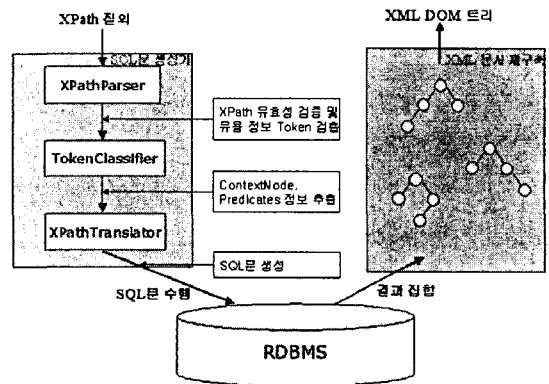


그림 3. XPath 질의 처리기 구조

본 논문에서는 질의 처리 단계 중 3번째 단계인 XPath구문을 SQL문으로 변환하는 질의 변환 단계와 결과 집합을 DOM 트리로서 생성하여 XML 문서를 재구축하는 부분을 수정하여 성능을 개선하였다.

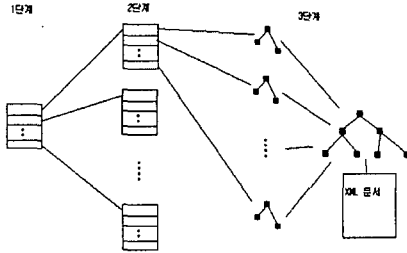


그림 5 이전 질의처리의 질의 변환 단계

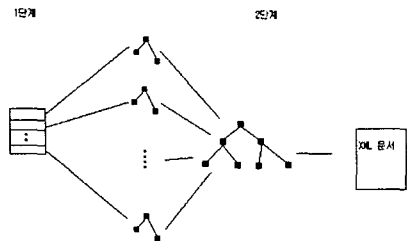


그림 6 개선된 질의처리의 질의 변환 단계

그림 4와 그림 5는 이전 질의 처리기와 개선된 질의 처리기의 질의 변환 단계를 나타낸 것이다. 본 논문에서는 첫 번째 단계와 두 번째 단계를 SQL:1999에서 지원하는 순환 질의를 이용해 한 단계로 처리했다.

질의 변환의 이해를 돕기 위해 XML 문서 내에서 address의 postcode값이 60325인 주문 사항의 주소를 확인하는 XPath질의인 `"/order//address[postcode='60325']"`를 사용한다.

XPath 질의 변환 단계에서는 토론 분류 단계에서 구성한 ContextNode 정보와 Predicate 정보를 활용하여 RDBMS에서 수행될 실질적인 SQL문으로 변환하는데, 이전 질의 처리기의 경우 먼저 반환되어야 하는 엘리먼트의 범위 ID를 얻는 질의를 생성하여 수행한다. 여기서 나온 결과를 가지고 범위 ID를 이용하여 범위 안에 있는 모든 엘리먼트, 애트리뷰트, 콘텐츠 등의 정보를 검색하고, 반환된 결과를 DOM 형태의 XML 데이터 조각으로 만든다.

반면에 개선된 질의 처리기는 SQL:1999의 새로운 시맨틱 중 하나인 순환 질의기법을 이용하여 다음과 같은 질의를 생성한다.

WITH

```
Q1 (SELECT el.id startId, el.lastId lastId
FROM helement e1, helement e2, content con1
WHERE e1.Pathexp LIKE '/order/%/address'
and e2.Pathexp LIKE '/order/%/address/postcode'
and e2.Id >= e1.Id and e2.lastId <= e1.lastId
and con1.htext ='60325' and e2.id = con1.elementId )
```

```
SELECT el.id, el.name, el.parentid, el.nodeorder,
attr.name, attr.value, attr.nodeorder,
con.nodeorder ,con.htext
FROM helement e1, helement e2, content con1, Q1
WHERE el.id = attr.elementId(+)
and el.id = con.elementId(+)
and el.id >= Q1.startId and el.id <= Q1.lastId
and el.xmldocumentId = Q1.xmldocumentId;
```

이 질의를 수행해서 얻는 결과물은 이전 질의 처리기의 질의 변환 단계 중 2단계에서 얻는 결과물과 동일하다. 이전 질의 처리기의 경우 1단계에서 나타나는 결과물을 분류하고 저장한 후 새로운 질의를 생성해서 다시 여러 번 질의를 해야만 원하는 엘리먼트, 애트리뷰트, 내용 정보를 얻을 수 있었으나 개선된 질의 처리기에서는 중간 질의 과정 없이 한번의 질의를 통하여 반복적으로 수행해야 하는 질의를 데이터베이스 내부적으로 해결하게 함으로써 오버헤드를 줄였다.

5. 결론 및 향후 연구 방향

본 논문에서는 DTD 독립적으로 저장하는 분할 방식 저장 시스템에서 XPath 질의를 통한 XML 데이터 검색을 가능하게 하는 질의 처리기를 새로운 SQL:1999 표준을 이용하여 설계함으로써 좀더 효율적으로 XML 데이터에 대한 검색질을 수행하도록 하였다.

또한 RDBMS의 결과값을 DOM 형식의 XML 문서로 생성 시킴으로써 타 응용 프로그램에서 바로 사용할 수 있게 되었다. 향후에는 본 저장 시스템을 가지고 SQL:1999의 강력한 객체-관계 DBMS 지원 능력을 활용하여 효율적인 XQuery 질의 처리를 수행할 수 있도록 할 예정이다.

6. 참고 문헌

- [1] J.Clark and S. DeRose, "XPath 1.0 : XML Path Language," W3C Recommendation, <http://www.w3.org/tr/XPath>, 1999.
- [2] D.Florescu and D.Kossmann, "Storing and Querying XML Data using an RDBMS," IEEE Data Engineering Bulletin 22(3), pp.27-24, 1999.
- [3] J. Shanmugasundaram, et al., "Relational Databases for Querying XML Documents : Limitations and Opportunities," Proc of the 25th VLDB Conf., 1999.
- [4] J. Shanmugasundaram, et al., "A General Technique for Querying XML Documents using a Relational Database System," SIGMOD Record, Vol. 30, No. 3, pp.20-26, 2001.
- [5] A. Eisenberg and J. Melton, "SQL:1999, Formerly Known as SQL3," SIGMOD Record, Vol. 28, No. 1, pp.131-133, 1999.
- [6] 고영기, 홍의경, "분할 저장 시스템에 적합한 XPath 질의 처리기 설계," 한국정보과학회 가을학술대회 논문집 29(2), pp.52-54, 2002.