XML 의 RDB 로의 맵핑과 효율적인 검색을 위한 색인 모델

김태희*, 김선경**
대구대학교 컴퓨터정보공학과*
대구대학교 컴퓨터·IT 공학부 교수**

A study on the XML Indexing model for Efficient Retrieval

Tae-hee Kim*, Sun-kyung Kim**
Dept. of Computer & Information Engineering, Daegu University*
School of Computer & Information Technology, Daegu University**

요 약

XML 문서의 관계형 데이터베이스로의 맵핑과, 데이터베이스의 성능을 향상시키기 위한 방안으로 내용 질의와 함께 XML 문서의 특성을 활용한 구조 검색 질의를 하기 위한 효율적인 색인 모델을 제안한다. 내용 색인과 구조 색인, 애트리뷰트 색인을 각각 구성하여 내용과 구조가 혼합된 효율적인 검색이 가능하도록 하였다. 검색의 효율성을 위해 노드 간의 위치 정보와 함께 독립적인 ID 를 부여하여 경로 테이블을 생성하여 질의의 효율을 높인다.

1. 서론

웹 상에서 디지털 정보를 교환하기 위한 표준으로 XML(eXtensible Markup Language)이 채택된 이후 각 방면에 대한 활용 방안이 꾸준히 연구되어 오고 있다. XML 은 데이터의 의미를 표현하는 언어로 데이터 분석이 가능하여 웹 상에 산재된 XML 데이터를 다양한 형태로 재구성할 수 있다.

데이터베이스적 측면에서 XML 은 내용을 표현하기 보다는 문서의 구조나 정보를 표현하기에 적합한 언어로, XML 전용 데이터베이스를 사용할 수도 있으나 일반적인 데이터베이스 시스템의 범용성과 지원하고 있는 제반 기술들을 사용할 수가 없다.

본 논문에서는 XML 문서의 검색 기능을 기존의 관계형 데이터베이스에서 구현하는 방법을 알아본다. XML 은 미리 정의된 스키마가 없고, 문서 자체에 데이터와 데이터 구조를 갖고 있기 때문에 기존의 SQL 등을 바로 적용 할 수가 없으므로, 먼저 XML 에 대해 질의 처리를 위한 스키마를 추출하고, DTD 에 포함된 엘리먼트와 구조 정보를 통해서 효율적인 검색을 위한 색인 구조를 만든 다음 질의를 처리하는 과정을 보인다.

2. 관련 연구

2.1 스키마 추출

XML 문서는 트리 형태의 그래프로 표현된다. XML 문서는 먼저 그래프 구조로 바꾸어 질의 처리를 위한 스키마를 추출하여야 한다. XML 에 대한 질의어와 질의 처리를 위한 스키마 추출에 관한 연구가 활발히 진행되고 있다. 그래프에서 각 엘리먼트는 노드로 표현 되고 두 노드 사이에는 간선이 존재하고 각 간선의 노드는 식별자를 가지고 방향성을 갖는다.[6]

1) 데이터 그래프
XML 문서의 모든 데이터가 표현되는 edge labled directed graph 로 정의된다. 노드로 부터 하위 노드로 방향성 있는 간선이 있고 간선의 레이블은 엘리먼트의 이름이 된다. 각 노드는 id 를 갖고 노드는 단순 객체 또는 복합 객체의 형태이다.

2) 스키마 그래프
XML 문서에 대한 데이터 그래프에서 깊이 우선 탐색 기법을 바탕으로 모든 경로가 단 한번만 표현될 수 있도록 만들어진 그래프이다.

3) 레이블 경로
데이터 그래프 혹은 스키마 그래프에서 한 노드에서 어떤 하위 노드로의 경로로 정의된다.

데이터 그래프를 통해 XML 문서의 구성을 쉽게 파악할 수 있고 XML 문서의 모든 요소가 표현되며 요소가 중복적으로 나타나는 경우도 있다.
XML 문서의 요소가 단 한번만 표현되는 스키마 그래프는 스키마 추출을 위해 기본적으로 필요한 그래프이다. 스키마 그래프는 레이블 경로의 빈도수에 따라 여러 개의 추출이 가능하므로 사용자 질의에 보다 효율적으로 처리할 수 있도록 레이블 경로 인덱싱을 사용한다.
레이블 경로는 스키마 그래프에서 루트 노드에서 리프 노드까지 깊이 우선 탐색 기법을 이용하여 하나씩 구하게 된다.

2.2 스키마 트리와 관계형 스키마간의 매핑

분해된 트리는 관계형 스키마로 맵핑 되어져야 한다. 스키마 트리와 관계형 데이터 테이블 사이의 매핑 정보는 XML 데이터를 저장하거나 사용자가 질의를 할 때, 질의에 대한 결과를 추출해내는 과정에서 필요하다. 매핑 정보는 XML 형태로 생성되기 때문에 일관성 있는 유지가 가능해진다.
XML 문서로부터 추출된 스키마 트리가 XML 문서와 관계형 매핑 정보를 생성하는데 사용되기 위해서는 객체-관계형 매핑 기법[5]을 적용한다. 스키마 트리를 객체 트리로 인식함으로써 클래스를 테이블로, 속성은 컬럼으로, 클래스 사이의 관계는 후보키/외래키 관계로 매핑된다.

이렇게 하면 스키마 트리의 각 엘리먼트는 클래스 타입과 속성 타입으로 나뉘어 지고 스키마 트리의 마지막 엘리먼트들은 리프 노드들이 속성 타입에 속하고 나머지 엘리먼트들은 클래스 타입에 속하게 된다.
스키마도 트리 형태의 계층 구조를 이루고 있다. 스키마 트리에서의 엘리먼트간의 부모/자식 관계는 두 엘리먼트들의 타입에 따라 결정 되어 지는데 두 엘리먼트들의 타입이 모두 클래스 타입이면 클래스-클래스 관계가 된다. 단일 값 속성은 클래스 테이블의 컬럼으로 매핑되거나 새로 생성된 테이블의 컬럼으로 매핑 되고, 다중값 속성은 별개의 테이블의 다중 튜플로 매핑된다.
XML 은 정의된 스키마가 없으나 문서 자체에 데이터와 데이터 구조를 갖고 있다. 그러므로 본 논문에서는 질의 처리를 위해서 먼저 스키마 추출을 하고 DTD 에 포함된 엘리먼트 구조를 통해 SQL 검색을 위한 색인 구조를 생성 시킨다.

3. XML 문서의 관계형 스키마로의 맵핑과 색인 모델

XML 스키마는 DTD 에 독립적인 형태를 사용하고, 구조적 맵핑을 위해 구조 정보를 나타내는 엘리먼트를 분석하여 관계 스키마를 정의한다. 스키마는 DTD 에 독립적인 형태로 변환하고, 엘리먼트의 추가와 삭제, 검색의 효율성을 위해 노드간의 위치 정보와 함께 독립적인 ID 를 부여한다.

3.1 XML 문서의 구조

XML 문서를 스키마를 생성시켜 저장한다.

```
<!-- ======= Start Entity Declaration ======= -->
<!-- ======= End Entity Declaration ======= -->

<!-- ======= Start Element Declaration ======= -->
<!ELEMENT papers (paper)+>
<!ELEMENT paper (style? , title , editor+ , author+ , summary)>
<!ELEMENT editor (society , date)>
<!ELEMENT society (#PCDATA)>
<!ELEMENT date (#PCDATA)>
<!ELEMENT author (lname+ , fname)>
<!ELEMENT lname (#PCDATA)>
<!ELEMENT fname (#PCDATA)>
<!ELEMENT summary (keyword)+>
<!ELEMENT keyword (#PCDATA)>
<!-- ======= End Element Declaration ======= -->
```

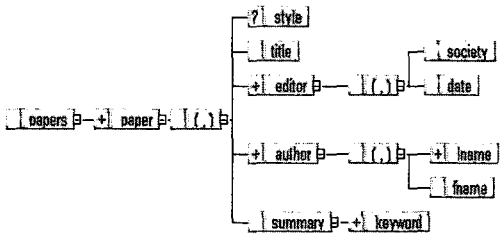

그림 1. XML 문서의 구조

엘리먼트는 EBNF(Extended Backus-Naur Form) 표기법에 따라 나타낸다.

3.2 색인 모델

XML 문서를 관계데이터베이스 스키마 구조로 변환하기 위하여 테이블을 생성하여야 한다. 먼저, 주어진 DTD 등을 분석하여 경로 정보 테이블을 구성한다. XML 문서를 왼쪽 자식, 오른쪽 형제의 구조를 갖는 이진 트리로 표현하고 각 엘리먼트에 임의의 고유한 식별자인 pid 를 부여하고 각 엘리먼트에 대해 엔트리로 매핑 한다.
구조적 검색 질의로의 확장을 위해 검색 경로를 엘리먼트 타입을 이용하여 최소화하기 하기 위하여 위치 정보를 내포하는 ID 를 부여한다. ID 의 부여는

0~9, A~Z, a~z 순으로 62 개의 문자를 사용하며 ASCII 코드의 순서를 따른다. 총 3844 개의 엘리먼트를 표현할 수 있다.

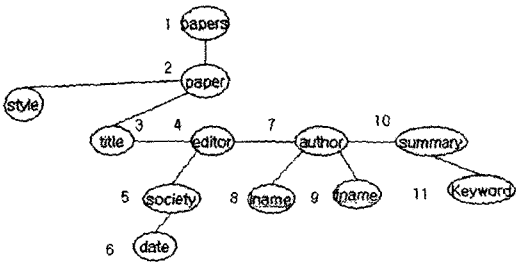

그림 2. XML 데이터 그래프

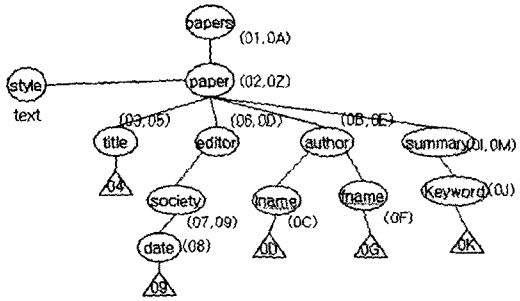

그림 3. 위치 정보가 부여된 그래프

3.3 테이블 구조

UID 테이블은 XML 문서를 실제적으로 저장하기 위한 테이블이다.

DTD 테이블은 XML 문서의 세부 정보를 나타내기 위한 테이블로 순서를 나타내는 pid, 데이터타입을 나타내는 type, 그리고 엘리먼트와 문서 내용을 가지고 있는 content 컬럼으로 구성된다.

NID 테이블은 전체 문서에 대한 일반적인 내용을 나타내는 테이블이고, 경로 정보를 나타내는 path 와 각 path 를 구분하기 위한 pathid, 시작 엘리먼트를 나타내는 sid, 종료 엘리먼트를 나타내는 eid 와 부모 엘리먼트를 나타내는 parid 로 구성된다.

DTD 테이블은 엘리먼트 정보를 담은 테이블이며, 각 엘리먼트의 목록 테이블이 EID 테이블이다.

표 1. 테이블의 구조

UID
docid(int)
docname(char)
content(char)

EID
docid(int)
element(char)
etid(varchar)

DTD
docid(int)
pid(varchar)
type(int)
content(char)

NID
docid(int)
path(varchar)
pathid(int)
sid(varchar)
eid(varchar)
parid(int)
pid(varchar)

3.4 검색

검색을 위한 질의는 트리 구조로 표현되며 각 엘리먼트는 노드로 맵핑 되어 식별자를 할당하고 인덱스를 생성시키고 주어진 경로식에 의해 XML 문서 트리를 방문하게 된다. 구조적 검색 시 조건 엘리먼트는 질의의 시작이 되는 부분이며, 검색 대상 엘리먼트는 질의의 최종 노드이며 방향 조건은 부모와 자식, 조상과 후손, 형제 등이 될 수 있다. 다음 예제에서 주어진 XML 문서와 맵핑된 테이블 형태에서 질의하는 과정을 보인다. XQL 형태의 질의는 관계 데이터베이스에서 지원되지 않으므로 SQL 형태의 경로 표현식으로 표현하여 질의한다.

1) papers 엘리먼트의 자식 엘리먼트 중의 하나인 summary 엘리먼트를 검색하라는 Xquery 질의 /papers//summary 에 대해 변환된 SQL 질의는 다음과 같이 수행된다.

검색 질의의 기준 엘리먼트는 papers 가 되고 검색 대상 엘리먼트는 summary 가 되어, papers 의 자식 엘리먼트이면서 주어진 질의와는 다른 경로상에 있는 엘리먼트의 접근은 필요하지 않게 된다.

```
use student
select DTD.content
from NID, DTD
where
NID.pid= DTD.pid
and NID.path LIKE '/003%/009'
FOR XML AUTO
```

XML_F52E2B61-18A1-11d1-B105-00805F49916B
1 <DTD content="This is a paper"/><DTD content="study."/>

2) Xquery //author/fname 에 대한 검색은 동일한 부모를 갖는 자식 중 같은 엘리먼트 명을 가지는 형제들의 자식은 부모 노드의 위치 값 parid를 기준으로 처리된다.

```
use student
select NID. sid. NID.eid
from DTD , NID
where DTD.pid=NID.pid and DTD.docid=NID.docid
and NID.docid=1 and NID.path LIKE '%/002%/001/005'
order by NID.sid
```

	sid	eid
1	0F	0H
2	0L	0N

위치 정보를 가진 pid 는 노드의 삽입시 논리적 위치 값이 변하지 않으므로 식별자가 될 수 있다. 노드의 삭제 시에도 pid 는 변하지 않지만 삭제된 노드를 참조하기 위해서는 노드의 parid 가 필요하다.

4. 결론

엘리먼트의 구조 정보와 위치 정보를 나타내는 색인 구조를 부여하여 XML 문서의 관계형 데이터베이스 스키마로의 맵핑에 있어서 구조적 특성을 고려하여 수행할 수 있는 문서 저장 구조를 제안하여 특정 엘리먼트에 대한 삭제 또는 새로운 엘리먼트의 삽입을 보다 용이하게 하였다. 삽입 또는 삭제가 발생하는 엘리먼트에 직접 연관된 엘리먼트에 대한 pid 만 적절하게 수정하게 되어 갱신에 대한 처리 시간을 줄일 수 있다. 삭제 또는 삽입된 노드시 대해 전체에 대한 pid 의 변경이

발생하지 않으므로 효율이고 XML 형태의 결과를 반환 받을 수 있다.
앞으로 다양한 데이터를 포함한 XML 문서로의 적용시 데이터의 증가로 인한 ID 부여 기법에 대한 지속적인 연구가 필요하다.

참고문헌

[1] 천윤우, 홍동권, "관계형 데이터베이스를 이용한 XQuery 전문 검색 ", 한국정보처리학회 추계학술발표대회 논문집, 제 10 권 2 호, 2003.11.

[2] 박경현, 이경휴, 류근호, "DTD 가 없는 XML 데이터의 효율적인 저장 기법", 정보처리학회 논문지 제 8-D 권 제 5 호, 2001.10.

[3] 김성완, "XML 문서에서의 순수 구조 질의에 대한 인덱싱 및 질의 처리", 한국정보과학회 추계학술 발표 논문집, 2002.

[4] 김성림, 윤용익, "XML 문서에서의 엘리먼트 정보를 이용한 스키마 추출 방법", 정보처리학회 논문지 제 9-D 권 제 3 호, 2002. 6.

[5] S. Malaika, "Using XML in Relational Database Applications", 15th Conf. on Data Engineering, Sydney, Australia, p.167, 1999.

[6] Roy Goldman, Jennifer Widom, "Data Guide : Enabling Query Formulation and Optimization in Semistructured Databases", In Proceeding of VLDB, 1997