

CXQuery (Chamois XML query language)의 XQuery 변환기 구현

이민영⁰, 용환승 이월영
이화여자대학교 서울기독대학교
mylee22@ewhain.net⁰, hsyong@ewha.ac.kr wylee@ewha.ac.kr.

Implementation of CXQuery to XQuery Converter

Minyoung Lee⁰, Wolyoung Lee, Hwanseung Yong
Ewha Institute of Science and Technology, Ewha Womans University

요 약

XML 문서로부터 정보를 검색하거나 추출하기 위해 쓰이는 질의 표준으로 XQuery 제안되었다. 하지만 사용자들이 XQuery를 사용하기 위해서는 XML문서의 구조를 알아야만 한다는 단점을 가지고 있다. 이에 문서 구조에 의존하지 않는 질의인 CXQuery가 제안되었지만 CXQuery가 XML을 파싱하여 DBMS에 저장한 뒤 DBMS에서 질의를 처리하기 때문에 XML문서를 직접 처리하지 못하고 한 단계 변환의 과정을 필요로 한다는 단점을 가지고 있다. 이 논문에서는, 사용자가 질의한 CXQuery를 그에 해당하는 XQuery로 변환하는 변환기를 개발함으로써 CXQuery의 문서 구조 독립적인 질의가 용이 하도록 하고 XML문서에 대해 직접적으로 질의 처리가 가능하도록 하였다.

1. 서 론

XML은 다양한 데이터 종류를 임의 형태로 쉽게 표현할 수 있도록 하는 데이터 모델이다. XML의 이러한 단순하고 유연한 특성 때문에 여러 응용 분야 즉, 전자상거래, 재정 서비스, 화학, 멀티미디어, 웹 서비스, 데이터 마이닝 등과 같은 분야에서 데이터를 교환하기 위한 표준으로 채택하고 있다. 이렇게 광범위 하게 사용되고 있는 XML 문서로부터 사용자들이 원하는 것을 검색할 수 있도록 하기 위해 많은 질의 언어들도 제안되었다. 그 중 W3C의 표준인 XQuery[1]는 지금까지 제안된 여러 가지 다른 질의 언어들을 기반으로 만들어져 가장 유연함에도 불구하고 질의가 XML 문서 구조에 절대적으로 의존하고 있기 때문에 문서 구조가 달라질 때마다 다른 질의 표현을 해야만 한다[2]. 이 점은 사용자를 불편하게 할 뿐만 아니라 문서 구조를 모르는 경우 거의 질의가 불가능하게 된다. 따라서 사용자가 문서의 구조를 모르더라도 질의가 가능한 CXQuery가 제안되었다[3].

CXQuery는 단지 검색하고자 하는 데이터 이름과 그 값만을 명시하고 데이터들 사이의 경로를 명시하지 않는 내용 기반 애드 혹 질의를 지원하도록 한다. 하지만 지금까지 CXQuery가 XML 문서를 파싱하여 DBMS에 저장한 뒤 DBMS에서 질의를 처리한다는 점에서 XML을 직접적인 쿼리를 처리하지 못하는 단점을 가지고 있다. 이에 사용자들이 편하기 사용할 수 있는 CXQuery로 질의한 것을 문서 구조에 기반한 XQuery로

바꾸어 줌으로써 XML 문서의 변환 없이 질의가 가능하도록 하였다.

본 논문의 구성은 다음과 같다. 2장에서는 CXQuery의 개념을 소개한다. 3장에서는 본 논문에서 제안하는 시스템의 구조와 방법에 설명하였다. 그리고 4장에서는 실험 및 결과를 5장에서는 향후 연구 방향과 결론에 대해서 살펴본다.

2. 관련 연구

XQuery는 XML이 다른 종류의 소스로부터 서로 다른 정보를 자유롭게 표현할 수 있다는 특징을 반영하기 위해 다양한 소스로부터 정보를 추출하고 해석하는 기능을 지원하도록 하고 있다. 또한 XML의 정보를 유연하게 질의 할 수 있도록 하기 위해 강력한 프로그래밍과 같은 기능을 지원하고 있다. 현재 Oracle 10g, MS sql2005 등의 상업 데이터베이스에서 XQuery가 지원된다.

그러나 대개의 질의 언어가 그러한 것처럼 XQuery도 같은 데이터를 가지고서도 수없이 많은 구조를 가질 수 있는 XML문서 구조에 너무 의존적인 질의 표현을 해야 하기 때문에 사용자들이 불편함을 느끼고 있다.

이러한 기존 언어들의 단점을 극복해 보고자 대개의 기존 XML 질의 언어가 XML 문서 구조를 따라 항해하는 방식으로 표현되는 경로 기반으로 질의하는 것에 반하여 문서 구조 독립적인 질의를 할 수 있도록 하는 CXQuery는 자연 언어에서와 비슷한 질의 표현식을 설계하였다. 예를 들어 아래와 같은 질의를 생각해 보자.

Q1: 1994년에 제작된 영화로서 genre가 action이고 'Jean Reno' 이라는 actor가 주연 한 영화의 제목은 무엇인가?

본 연구는 한국과학재단 목적기초연구(R01-2003-000-10395-0) 지원으로 수행되었음

위와 같은 질의를 하기 위해서는 사용자는 적어도 검색을 원하는 데이터 이름과 그 값은 알고 있어야 한다. CXQuery는 사용자가 질의하고자 하는 데이터 이름과 값만을 안다면 구조에 대해서는 고민하지 않고 데이터의 형태 (Element or Attribute)나 데이터의 경로에 상관없이 이들 모든 문서에 대해 다음과 같은 똑 같은 하나의 질의 표현으로 질의가 가능하도록 하였다.

genre=" action" AND actor=" Jean Reno"
따라서 사용자는 질의에 대해 문서의 구조에 대한 고려 없이 편하고 쉽게 검색 가능하다.

그러나 CXQuery의 질의 처리기는 XML문서를 파싱하여 기존의 데이터베이스에 저장한 후에 질의 처리하는 기법을 사용하기 때문에 기존 DBMS에 의존적이고 XML문서의 직접적인 처리가 어렵다는 단점을 갖는다.

따라서 본 연구에서는 XML 문서를 직접 처리할 수 있도록 CXQuery를 XQuery로 변환하는 변환기를 구현하였다.

3. CXQuery 변환기의 설계 및 구현

제안된 시스템은 아래와 같은 순서로 질의를 처리한다.

- 1) 사용자는 User interface로 CXQuery 입력한다.
- 2) CXQuery를 Xquery로 변환하여 User interface에 만들어진 XQuery를 사용자에게 반환
- 3) 만들어진 Xquery로부터 사용자가 원하는 쿼리 선택
- 4) 선택된 쿼리를 실행 (다양한 DBMS 사용 가능)

하나의 CXQuery는 문서 구조에 따라 여러 개의 XQuery로 변환이 될 수 있다. 사용자가 원하는 질의는 만들어진 XQueries 중에서 단 몇 개에만 불과할 것이기 때문에 사용자가 원하는 질의를 찾아내는 방법으로서 사용자가 직접 원하는 쿼리를 선택하도록 하였다.

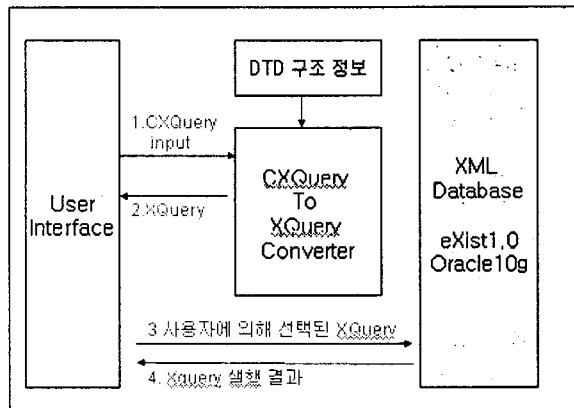


그림 1 변환기를 사용한 XML DBMS 시스템 구조도

3.1 CXQuery를 XQuery로 변환

그림 3에서 보여주는 것처럼 CXQuery를 XQuery 로

변환할 때 XML 문서의 구조를 아는 경우와 문서의 구조를 모르는 경우를 고려해야 한다.

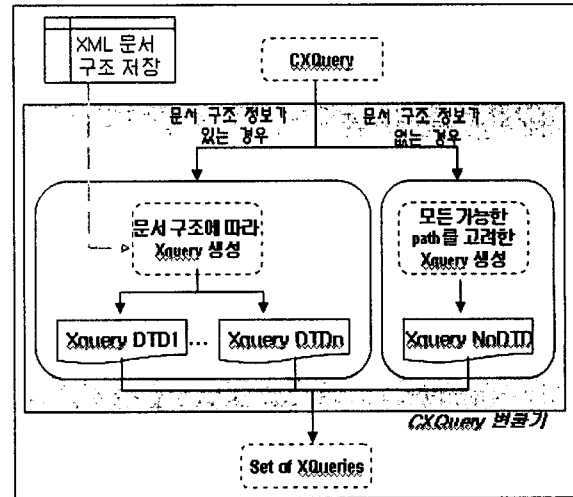


그림 2 CXQuery To XQuery Converter

3.1.1 XML 문서의 구조를 모르는 경우

CXQuery를 주어줄 때 사용자는 데이터 이름과 값을 주어 질의를 한다. 예를 들어, 장르가 액션이라고 할 때 사용자는 genre=" action" 이라고 조건을 쓰게 된다. 이 조건을 XQuery로 바꾸어 주기 위해서는 먼저 XML 문서에서 이 조건이 쓰일 수 있는 모든 가능한 경로를 고려해 주어야 한다[4]. 위 조건이 문서에서 쓰일 수 있는 모든 가능한 경로는 표 1과 같다.

표 1 genre의 모든 가능한 경로들

문서	데이터 경로	쿼리
문서 1	데이터 이름이 element이고 이 element가 값을 가지는 경우	//genre=" action"
문서 2	데이터 이름이 attribute인 경우	//@genre=" action"
문서 3	데이터 이름이 element이고 이 element의 자식 element가 값을 가지는 경우	//genre//*" action"
문서 4	데이터 이름이 element이고 이 element의 자식 element의 attribute가 값을 가지는 경우	//genre//@*" action"

위의 모든 가능한 구조를 토대로 위의 조건은 다음과 같은 XQuery 조건으로 변환 가능하다.

//genre=" action" OR //@genre=" action" OR //genre//*" action" OR //genre//@*" action"

이러한 조건에 의해 Q1을 CXQuery에서 XQuery로 변환한 것의 결과가 다음과 같다.

표 2 Q1에 대한 CXQuery와 XQuery 질의 표현

CXQuery	XQuery
For \$c in doc()	For \$c in doc()
Where genre=" action	Where (\$c//genre=" action" OR \$c //genre//!=" action" OR \$c //@genre=" action" OR \$c//genre//@*=" action")
AND actor=" Jean Reno"	AND (\$c//actor=" Jean Reno" OR \$c//actor//!=" Jean Reno" OR \$c //@actor=" Jean Reno" OR \$c//actor//@*=" Jean Reno")
Return title	Return \$c//title

3.1.2 XML 문서의 구조를 아는 경우

XML 문서의 구조를 아는 경우에는 CXQuery에 적힌 데이터 이름에 해당하는 노드를 찾아 완전한 경로를 적어 줌으로써 문서의 구조에 맞춰 XQuery로 변환하여 주면 된다. 문서의 구조가 다를 경우 데이터 경로도 달라지기 때문에 DTD 하나당 한 개의 XQuery가 생성된다.

```
<ELEMENT movies (movie+)>
<ELEMENT movie (year, country,
genre, title, director, actor)>
<ELEMENT country (#PCDATA)>
<ELEMENT genre (#PCDATA)>
<ELEMENT title (#PCDATA)>
<ELEMENT director (#PCDATA)>
<ELEMENT actor (#PCDATA)>
<!ATTLIST movie year
CDATA #REQUIRED>
```

type	path	name
ELE	/	movies
ELE	/movies	movie
ATT	/movies/movie	Year
ELE	/movies/movie	Country
ELE	/movies/movie	Genre
ELE	/movies/movie	Title
ELE	/movies/movie	Director
ELE	/movies/movie	actor

그림 3 XML 문서 구조와 구조 정보의 저장

그림 3는 데이터들이 모두 중첩 되지 않은 가장 단순한 구조를 보이고 있다. 이 문서의 구조 정보를 각 노드의 타입, 그 노드까지 이르는 경로, 노드의 이름을 따로 저장한 모습을 볼 수 있다.

이렇게 문서의 구조를 알고 있다면 title 의 경로 /movies/movie/title, genre의 경로 /movies/movie/genre 와 action의 경로 /movies/movie/action을 구할 수 있다. 다양한 문서 구조에 따라 표2와 같이 다양한 구조에 따른 XQuery가 생성될 수 있다. 이 정보를 이용해 CXQuery를 수정해 준다.

- 1) 노드간의 동일한 패스를 구해 For 문 수정
=> For \$c in /movies/movie
- 2) CXQuery의 각 노드 패스를 수정
=> Where \$c/genre=" action" AND
 \$c/actor=" Jean Reno"
=> Return \$c/title

표3은 다양한 문서 구조에 따라 생성될 수 있는 XQuery의 예이다.

표3 문서 구조에 따른 XQuery

문서 구조 1	For \$c in /movies/movie Where \$c/genre=" action" AND \$c/actor=" Jean Reno" Return \$c/title
문서 구조 2	For \$c in /movies Where \$c/movie/genre=" action" AND \$c/movie/genre/actor=" Jean Reno" Return \$c/ title
문서 구조 3	For \$c in /movie Where \$c/movie/@genre=" action" AND \$c/movie/title/@actor=" Jean Reno" Return \$c/movie/title
문서 구조 4	For \$c in /movies Where \$c/movie/@genre=" action" AND \$c/movie/actor=" Jean Reno" Return \$c/movie/title

4. 실험 및 결과

본 연구에서는 JAVA(J2SE v1.4)로 구현하였고 XML 문서를 저장하는 데이터베이스로 eXist(v1.0B)[5] 사용하였다.

시스템 실행 결과 CXQuery를 XQuery로 변환하는데 걸리는 시간은 무시할 정도로 적었으며 문서 구조에 따라 만들어질 수 있는 모든 경로를 따져 XQuery가 변환되어 사용자가 원하는 질의를 선택하도록 하므로써, 사용자가 원하는 질의 결과를 모두 구할 수 있었다.

5. 결론

본 논문에서는 사용자가 문서의 구조를 모르더라도 데이터 이름과 값만으로도 질의가 가능한 CXQuery를 XML문서에 직접 질의가 가능한 XQuery로 변환하는 프로그램을 개발하였다. 이로 인해 사용자에게는 편이를 제공하고 사용자가 만들어진 XQuery들로부터 자신의 원하는 질의를 선택하여 선택된 질의만 수행함으로써 CXQuery의 모호성을 줄일 수 있었다.

참고 문헌

- [1] W3C Consortium, XQuery1.0: An XML Query Language, W3C Working Draft, Jun. 2001
- [2] W3C Consortium, XQuery1.0 Requirements, W3C Working Draft, 15 Feb. 2001
- [3] 이월영, XML 데이터베이스에서 문서 구조 독립적인 질의 처리 기법, 이화여대 과학기술대학원 박사학위논문, 2004. 8
- [4] W3C Consortium, XML Path Language(XPath) Version 1.0, W3C Recommendation, 16 Nov.1999
- [5] eXist, <http://www.exist-db.org>