

XQuery 작성기 설계 및 구현

김태권

강남대학교 컴퓨터미디어공학부
ktg@kangnam.ac.kr

Tae Gwon Kim
School of Computer Media, Kangnam University

요 약

XML은 관계형 데이터는 물론 구조화 또는 반구조화된 데이터를 효과적으로 조직화하여 표현할 수가 있다. XQuery는 구조화된 XML 데이터를 대상으로 필요한 정보를 검색하는 질의어이다. 평면적인 테이블 형태의 SQL과는 달리 XQuery는 데이터의 내부구조 정보없이 질의어를 작성하는데 어려움이 따른다. 이 논문은 내부적으로 구조화된 XML 데이터에서 필요한 정보를 검색하는 검색언어 XQuery 질의를 효과적으로 작성할 수 있도록 질의 대상이 되는 XML 데이터 구조를 트리 형태로 보여주고, 필요한 경로식을 효과적으로 지정함으로써 질의어를 보다 쉽게 작성하도록 도와주는 XQuery 작성기를 설계하고 구현한다.

1. XML 데이터

XML[1]은 사람이 이해하기 쉽고 기계가 다루기 쉬운 구조를 제공하여 웹 기술상에 있어서 혁신적인 변화, 안전한 전자상거래 구축, 새로운 분산 어플리케이션 등에서 관심이 많으며, 고정된 태그가 없고 새로운 태그를 필요에 따라 언제든지 만들어 사용할 수 있다. XML은 태그와 속성(attribute)을 이용해 스키마의 정의 없이도 데이터베이스처럼 사용할 수 있다. XML의 태그는 내용의 의미를 기술할 뿐만 아니라 XSL을 이용하여 프리젠테이션 양식을 제공함으로써 하나의 문서 하나는 여러 개의 프리젠테이션 문서(XSL)에 의하여 다양한 역할을 가질 수 있다.

XML 문서는 크게 두 부분으로 구성되어 있다. 즉 XML 선언 부분으로서 DTD를 정의하는 프롤로그와 실제 XML 문서의 내용을 입력하는 도큐먼트 인스턴스 부분으로 구성되어 있다. XML 문서의 스키마를 의미하는 DTD는 문서 내에서 사용할 태그들을 정의하기 위한 일련의 문서 구조에 대한 규칙이다. DTD는 문서 안에서 어떤 태그를 쓸 수 있고, 태그들이 문서 안에서 어떤 순서로 나타내야 하며, 다른 문서들 속에서 어떤 태그들이 나타날 수 있는지, 어떤 태그가 속성(attributes)을 갖는지 문서의 구조를 정의한다. 문서가 마크업되는 방식에 대한 정의하는 XML Schemaf[2], XML 문법을 이용해서 스키마를 기술하는 XML-Data[3], 개선된 자원기술 프레임워크(Resource Description Framework)의 문법을 이용해서 문서구조를 기술하는 DCD(Document Content Description for XML) 등이 있다.

2. XQuery

XML 일반적으로 문서들은 규칙적이지 못한 구조를 가지고 있으며, 깊이 중첩되어(nest) 있고, 상대적으로 간단한 데이터 타입을 사용하고 있으며, 나타나는 순서에 중요한 의미를 부여하고 있다. 반면에 관계형 데이터베이스에서는 매우 규칙적인 구조를 가지며, 상대적으로 평면적인 단순한 구조이고, 매우 복잡한 데이터 타입을 사용하며, 순서에는 그렇게 의미를 부여하지 않는다. XML은 편집이 가능하고, 쉽게 파싱이 가능하며, 구조화 또는 반구조화된 어떠한 형태의 문서도 표현할 수가 있

어 정보교환을 위한 포맷으로 유용하게 사용될 수가 있다. XML이 이러한 융통성에 의하여 문서와 데이터베이스 사이의 차이가 사라지고 있으며, XML이 정보의 다양한 형태를 쉽게 표현함에 따라 정보의 표현이 XML로 통합되는 효과가 나타나고 있다.

XML이 정보 교환을 위한 일반적인 포맷으로 그 잠재성을 실현하기 위해서는 XML에 못지 않게 융통성 있는 질의 언어가 요구된다. 문서 내에서 검색을 하기 위하여 질의어는 문서 일부의 순서와 중첩된 구조를 유지할 수 있어야 한다. 데이터베이스에 대한 검색은 선택, 프로젝션, 조인, 그룹화와 같은 연산을 제공한다. 문서에 대한 검색은 XML Schema의 규정에 있는 모든 정보의 구조를 다룰 수 있어야 하며 정보를 또 다른 형태의 구조로도 변형할 수 있어야 한다. 검색언어는 또한 간결하고 해독성이 있어야 하고, 어떠한 형태의 XML 정보에 대하여 검색을 할 수 있을 정도로 융통성이 있어야 한다. XQuery는 여러 검색언어에서 필요한 특성들을 빌려와서 충분한 역할을 할 수 있도록 설계되었다.

XQuery[4]는 여러 종류의 수식을 제공하며, 검색어는 여러 형태의 형식을 가진다. 일반적으로 XQuery의 수식의 다양한 형식이 내부적으로 거의 아무런 제약없이 중첩될 수도 있어, 부분질의(subquery)와 같은 개념이 XQuery에게는 매우 자연스럽다. XQuery 질의의 입력과 출력은 XML 문서, XML 문서의 부분, 또는 XML 문서들의 집합으로 볼 수 있다. XQuery 검색에 대한 이러한 입력과 출력을 XML 질의 데이터 모델의 인스턴스로 간주할 수 있고, 이러한 XML 질의를 위한 데이터 모델은 W3C XML Query Working Group에 의하여 개발되고 있으며, 노드들의 트리로 모델링되는 XPath[5] 규정의 데이터 모델을 좀 더 개량한 것으로 볼 수 있다.

문서의 일부분이나 문서들의 집합은 서로 공통된 루트(root)가 없으며, 원소 노드, 애트리뷰트 노드, 텍스트 노드들을 포함한 다양한 노드 타입들의 숲(forest)로 모델링된다. XQuery 수식의 기본적인 형태는 경로식, 원소 생성자, FLWR 수식, 연산자, 조건수식, 한정자, 변수 바인딩 등으로 구성된다. XQuery의 경로식은 XPath의 구문을 바탕으로 생략된 형태의 구문을 사용하여 계층구조의 여러 노드들을 이동하는 방법을 제공한

다. 경로식의 결과는 경로식의 패턴을 만족하는 노드들과 그들의 자손으로 이루어지는 트리들의 집합이다. XPath 구문은 XSLT, XPointer와 같은 여러 XML을 기반으로 하는 응용프로그램에서 또한 사용된다. XPath처럼 XQuery 경로식은 일련의 연속적인 계층구조의 스텝(step)로 구성된다. 각 스텝은 특정한 방향으로 문서를 향해하는 이동으로 표현된다. 각 스텝은 주어진 조건을 만족하지 않는 노드들은 제거하는 산술문(predicate)을 적용할 수 있다. 각 스텝의 결과는 다음 스텝을 위한 출발점 역할을 하는 노드들의 집합이다.

경로식은 document와 같은 함수를 사용하여 특정의 노드를 지정함으로써 시작된다. XQuery는 '/' 또는 '//로 시작되는 경로식을 가질 수가 있으며, 암시적인 루트 노드를 의미한다. 다음의 예들은 경로식에 대한 간단한 설명이다.

```
document("book.xml")/chapter[2]/figure[caption="Frogs"]
```

위의 경로식은 세 스텝으로 구성되어 있는데, 첫 번째 스텝은 대상이 되는 XML의 파일은 "book.xml"이고, document라는 함수는 이 문서의 루트를 반환한다. 두 번째 스텝은 루트노드의 자식인 노드들 중에서 chapter라는 원소를 찾고 그 중에서도 2 번째의 원소를 표현한다. 세 번째 스텝은 그 chapter 이내에서 모든 후손 원소들 중에서 "Frogs"라는 caption 원소를 가지는 figure 원소를 표현하고 있다.

가장 핵심적인 수식은 FLWR는 For, Let, Where, Return 절에 의하여 만들어진다. SQL에서와 같이 이 절들은 지정된 순서로 나타난다. FLWR 수식은 변수들에게 값을 바인딩함으로써 시작되고 이 변수들은 결과를 생성하기 위하여 사용된다. 먼저 For 절은 변수들의 바인딩을 생성하는 절이다.

For 절에서 지정된 변수는 수식과 관련이 되고, 일반적으로 이 수식들의 각각은 노드들의 리스트를 반환되고, For 절의 결과는 튜플들의 집합으로 생각할 수 있다. 그리고 여러 For 절에 의하여 바인딩된 변수들은 각각에 대한 cross-product를 의미한다.

Let 절은 For 절 다음에서 추가적인 변수의 바인딩을 하기 위하여 사용되는데, 일반적으로 For 절에서 정의된 변수를 사용하여 새로운 변수를 지정하기 위하여 사용된다. FLWR 수식은 여러 For 절과 Let 절을 가질 수가 있으며 SQL의 FROM 절에 대응된다.

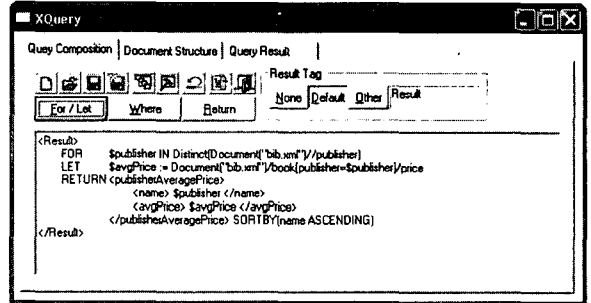
Where 절은 For 절과 Let 절에서 바인딩된 튜플들의 각각에 Where 절에서 필터링되는데 SQL의 Where 절에 대응된다. Where 절의 조건을 만족하는 튜플들만 Return 절에서 원소를 생성하기 위하여 사용될 수 있다. Where 절은 여러 산술문인 AND, OR, NOT 등으로 연결되어 사용된다.

Return 절은 FLWR 수식의 결과를 생성하여 반환한다. 그 결과는 노드, 순서가 있는 노드들의 리스트, 또는 기본적인 값이 될 수가 있다. Return 절은 For 절과 Let 절에서 바인딩된 튜플들에 대하여 결과의 원소를 생성하기 위하여 한번씩 실행된다. 이 튜플들 사이에 순서가 있으면 Return 절은 순서에 따라 각 튜플들에 의하여 실행되고, 결과도 그 순서에 따라 결과 문서에 저장된다.

3. XQuery의 설계 및 구현

XQuery는 관계형 데이터베이스 질의어 SQL처럼 구문은 아주 간결하며 단순하고 구조적으로 되어 있다. 그런데 문체는 XML 문서 자체가 내부적으로 구조적으로 조직화되어 있고,

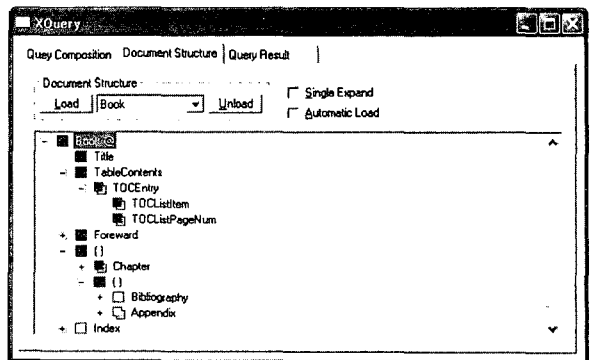
문서마다 그 구조가 서로 다를 수 있으므로 질의를 작성하는 많은 어려움이 따른다. 특히 문서의 계층구조에 따른 경로식을 작성하기가 매우 어렵다. XML 문서에 대한 검색을 효과적으로 하기 위해서는 문서의 구조를 확실하게 파악하여 경로식을 정확하게 작성하는데 있다. 이러한 어려움을 해소하기 위하여 XQuery의 질의를 작성하는 도움이 될 XQuery 작성기를 설계하고 구현하였다.



[그림 1] XQuery 작성기

XQuery 작성기는 XML 문서에 대한 질의를 효과적으로 작성할 수 있도록 XML 문서에 대한 계층적인 구조를 미리 로드하여 경로식이 필요로 할 때마다 필요한 경로를 체크하게 함으로써 질의 작성을 도운다. 질의 작성기는 [그림 1]과 같이 질의 작성, 문서 구조, 질의 결과의 3 개의 탭과 여러 버튼들로 구성되어 있다. 질의 작성 탭은 현재 작성하고 있는 질의를 보여준다. 문서 구조는 질의의 대상이 되는 문서의 구조를 로드하도록 지원한다. 필요에 따라 여러 개의 구조를 동시에 로드할 수 있고, 원하는 구조를 골라서 이용할 수 있다. 질의 결과는 작성된 질의를 실행한 결과를 보여준다. 아랫부분은 편집기이기 때문에 원하는 대로 편집을 할 수 있다.

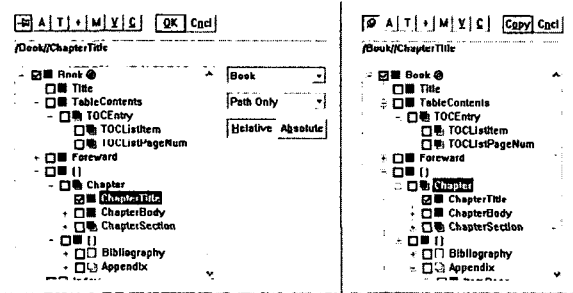
버튼은 2줄로 구성되어 있다. 윗줄의 버튼은 일반적인 기능을 하고, 아랫줄의 버튼은 각각의 대화상자를 띄운다. 먼저 윗줄의 버튼을 먼저 설명한다. 첫 번째 버튼은 새로운 질의를 작성하게 초기화시키고, 두 번째 버튼은 이미 작성된 질의를 오픈하게 하고, 세 번째 버튼은 작성된 질의를 저장하고, 네 번째 버튼은 새로운 이름으로 저장하게 하고, 다섯 번째 버튼은 문서의 구조를 보여주고, 여섯 번째 버튼은 도구상자를 보여주고, 일곱 번째 버튼은 수정된 질의를 undo시키고, 여덟 번째 버튼은 작성된 질의를 실행시키고, 아홉 번째 버튼은 실행을



[그림 2] 문서의 구조

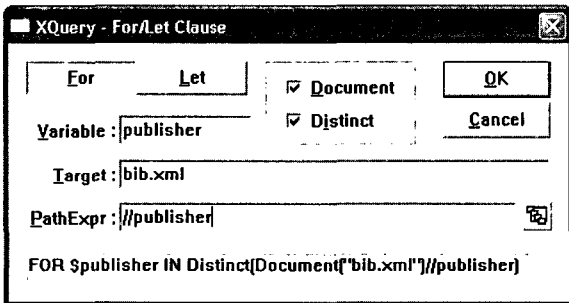
में출다.

문서의 구조는 [그림 2]처럼 그래픽으로 보여준다. 이 구조는 XPath를 작성하는 기본적인 정보로 활용될 수 있다. 문서 구조는 근본적으로 트리로 자식들은 숨기거나 필요로 하면 확장하여 볼 수도 있다. 여러 개의 문서 구조를 동시에 로드하여 원하는 구조를 선택하여 볼 수도 있고, 더 이상 필요로 하지 않을 경우는 메모리에서 내릴 수도 있다.



[그림 3] 경로식

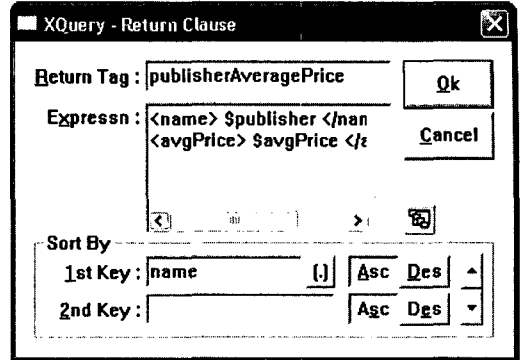
다양한 경로식을 효과적으로 작성하도록 도와주는 대화상자는 [그림 3]에서 보여준다. 둘 다 기능은 같지만 사용하는 방법이 조금 다르다. 경로식이 하나면 필요할 경우에는 원본과 같은 대화상자를 이용하여 경로식을 트리 구조 상에서 필요로 하는 원소만 체크함으로써 아주 편리하게 작성할 수가 있다. 작성이 끝나면 OK 버튼을 누르면 경로식이 편집기 속의 커서 위치로 자동적으로 삽입되며 대화상자는 없어진다. 그러나 경로식을 여러 번 필요로 하는 경우에 첫 번째의 핀 버튼을 누르면 대화상자가 오른쪽 그림처럼 고정되고 OK 버튼은 COPY 버튼으로 변한다. 이러한 상황에서는 경로식이 작성된 후에 클립보드에 복사되고 대화상자도 그대로 유지된다. 경로식이 필요한 곳에 커서를 이동하고 붙이기를 하면 경로식을 얻을 수 있다. 대화상자가 계속 유지되기 때문에 얼마든지 반복할 수 있다.



[그림 4] For/Let 절 대화상자

For/Let 절을 효과적으로 작성할 수 있도록 도우는 대화상자는 [그림 4]에서 보여준다. 라디오 버튼을 사용하여 For 절 또는 Let 절을 설정하고, 변수 이름, 대상 XML 파일, 경로식을 입력하면 아래에 편집되는 그 내용을 확인할 수 있다. 경로식이 필요로 하면 오른쪽 하단의 버튼을 누르면 [그림 3]과 같은 대화상자를 이용하여 경로식을 작성하면 그 결과의 경로식이 자동적으로 입력된다.

Return 절을 효과적으로 작성하도록 도와주는 대화상자는 [그림 5]에서 보여준다. 원소 생성자를 위한 태그 이름과 원소



[그림 5] Return 절 대화상자

의 내용을 편집기를 사용하여 채운다. 경로식이 필요로 하면 오른쪽 중간의 경로식 버튼을 누르면 편집기에 적성한 경로식이 자동으로 입력된다.

그 밖에도 산술문, 필터, 조건문, 한정연산자, Befor/After를 위한 도구상자가 있고, 이를 이용하여 다양한 대화상자가 나타나고 원하는 질의를 효과적으로 작성할 수 있도록 도와준다.

4. 결론

XQuery는 XML 문서를 대상으로 필요한 정보를 검색하는 언어이다. 이 검색 언어는 SQL처럼 간결하고 명료하게 구조화되어 있어 쉽게 사용할 수 있다. 그런데 질의 대상이 되는 XML 문서는 내부적으로 계층구조를 형성하고 있어 접근하기가 용이하지 않을뿐더러 이로 말미암아 XQuery 작성에도 어려움이 따른다. 이러한 문제점을 해결하기 위하여 XQuery 작성기는 검색대상이 되는 문서의 구조를 미리 로드하여 사용자에게 트리 구조의 그래픽으로 보여주고 원하는 부분만 체크하게 함으로써 경로식을 효과적으로 작성하게 함으로써 XML 문서에 대한 질의어 XQuery를 효과적으로 작성할 수 있었다.

5. 참고문헌

- [1] Tim Bray, Jean Paoli, C. M. Sperberg-McQueen, Eve Maler, François Yergeau, "Extensible Markup Language (XML) 1.0 (Third Edition)", <http://www.w3.org/XML/>, 2004.
- [2] David C. Fallside, "XML Schema Part 0: Primer", <http://www.w3.org/TR/xmlschema-0/>, 2001.
- [3] Andrew Layman, Edward Jung, Eve Maler, Henry S. Thompson, Jean Paoli, John Tigue, Norbert H. Mikula, Steve De Rose, "XML-Data", <http://www.w3.org/TR/1998/NOTE-XML-data-0105/>, 1998.
- [4] Scott Boag, Don Chamberlin, Mary F. Fernandez, Daniela Florescu, Jonathan Robie, Jérôme Siméon, "XQuery 1.0: An XML Query Language", <http://www.w3.org/TR/xquery/>, 2004.
- [5] James Clark, Steve DeRose, "XML Path Language (XPath) Version 1.0", <http://www.w3.org/TR/xpath/>, 1999.