제21회 한국정보처리학회 춘계학술발표대회 논문집 제11권 제1호 (2004. 5)

XQuery 에서의 group by 도입을 통한 효율적 질의 표현기법

조혜영 O, 이민수
이화여자대학교 컴퓨터학과
e-mail : hycho O@ewha.ac.kr, mlee@ewha.ac.kr

Efficient expression of queries by extending XQuery with group by

Hyeyoung Cho O, Minsoo Lee
Dept. of Computer Science & Engineering, Ewha Womans University

요 약

XML(eXtensible Markup Language)은 용이한 이식성, 정보 표현의 유연성 및 폭넓은 확장성 등의 장점을 가지고 있어 웹상에서의 데이터 표현, 상호교환, 검색을 위한 포맷으로 널리 사용되고 있다. XML 문서의 검색을 위해 다양한 질의 언어들이 제안되었는데 그 중 XQuery 는 사실상 XML 질의 언어의 표준으로 자리잡았다. 본 논문에서는 현재 XQuery 구문에서 그 기능이 제공되고 있지는 않지만 일반적인 SQL 에서처럼 데이터의 재구성과 집계처리를 위한 그룹화를 쉽게 해주는 명시적인 group by 도입을 통한 효율적인 질의 표현 기법을 제안하고자 한다.

1. 서론

XML(eXtensible Markup Language)은 용이한 이식성, 정보 표현의 유연성 및 폭넓은 확장성 등의 장점을 가지고 있어 웹상에서의 데이터 표현, 상호교환을 위한 포맷으로 널리 사용되고 있다. 이러한 XML 문서의 광범위한 사용과 XML 기반 정보의 증가에 따라 XML 문서의 효율적인 검색과 관련된 연구 역시 활발히 진행되고 있다.

XML 문서의 검색을 위해서 XQL, XML-QL, Quilt[1], XPath[2], XQuery[3] 등과 같은 다양한 질의 언어들이 제안되었다. 이 중에서 XQuery 는 여러 가지 형태의 XML 데이터 소스에 적용할 수 있도록 설계되어 사실상 XML 질의 언어의 표준으로 자리잡았다.

본 논문에서는 현재 XQuery 구문에서 그 기능이 제공되고 있지는 않지만 일반적인 SQL 에서처럼 데이터의 재구성과 집계함수 처리에 유용한 그룹화를 쉽게 해주는 명시적인 **group by** 도입을 통한 효율적인 질의 표현 기법을 모색하고자 한다.

본 논문의 구성은 다음과 같다. 먼저 XQuery 및 그룹화 처리와 관련된 연구들에 대해 소개하고, 다음으로 XQuery 에서의 명시적인 **group by** 구성을 이용한 그룹화 처리가 필요한 이유와 **group by** 도입을 통한 XQuery 의 확장 방안을 제시하고, 마지막으로 결론 및 향후과제를 살펴본다.

2. XQuery

XQuery 는 W3C 의 XML Query Working Group 에서 권고하는 질의 언어로서 Quilt 를 근간으로 하여 다른 언어들이 가지고 있는 유용한 기능들을 추가하였으며, 구조적(structured) 혹은 반-구조적(semi-structured) 문서, 관계형 데이터베이스(relational database), 오브젝트 저장소(object repository) 등 다양한 데이터 소스로부터 가져온 정보를 고유한 구조를 가진 질의 결과로 구성할 수 있다.

질의 표현식은 SQL 의 Select-From-Where 에 대응하는 이른바 FLWR(For-Let-Where-Return)식이라 부르는 구문에 따라 구성되며, 구조적 질의어인 XPath 식을 이용하여 XML 문서의 계층적 경로를 표현한다. 모든 데이터 형식에 대해 order by 를 사용한 정렬 및 avg(), count() 등과 같은 집계함수(aggregate function), 엘리먼트 생성(element constructor), deep-equal 에 의한 구조적 일치성 비교, distinct-values 를 사용한 중복제거, if…then…else 조건 표현식(conditional expression), some/every… satisfies… 정량 표현식(quantified expression) 등 다양한 기능을 지원하고 사용자 정의 데이터 타입 및 사용자 정의 함수도 지원하나 아직까지 명시적인 **group by** 지원에 대해서는 언급된 바가 없다.

3. 그룹화 처리(groupwise processing)

그룹화는 데이터를 재구성하거나 집계함수를 처리하는 경우 주로 이용되며, 관계형 데이터베이스에 있어 중요한 연산 중 하나이다.

데이터 웨어하우징의 맥락에서, 데이터 웨어하우징 어플리케이션을 위한 튜플들의 그룹화 처리에 대한 연구들이 있었다. 핵심 아이디어는 직관적으로 데이터들을 튜플들의 그룹으로 분할하고 이들에 대해 반복적인 계산을 수행하는 질의를 처리하는 것이다. 이를 위해 그룹화와 집계처리를 목적으로 고안된 새로운 연산자에 따르도록 기존의 SQL 을 확장하거나[4], 그룹방식 처리에 해당하는 질의인지 확인하여 구문적으로 SQL 질의를 처리하는 대안적인 접근을 채택하기도 하였다[5].

이처럼 그룹화는 관계형 맥락에서는 이해되고 있지만, 복잡한 구조를 가지는 XML 에서는 그룹화의 중요성이 더 큼에 불구하고, 그룹화를 지정하고 구현하기가 쉽지 않다. XML 문서의 일부 애트리뷰트나 하위 엘리먼트가 존재하지 않거나 하위 엘리먼트가 반복되는 등의 구조적 유연성에 의한 영향을 많이 받기 때문이다.

XML 에서의 그룹화 처리를 위한 연산자를 제안하고, XML 과 관계형 데이터 모델간의 기본적 차이에 대한 통찰력을 제공한 연구가 있었다[6]. 이 연구에서는 관계형 엔진이 단일 튜플이 아닌 튜플들의 집합에 대한 바인딩 변수를 지원하도록 하는데 주안점을 두고, 기존 관계형 엔진에 고르게 통합될 수 있는 GApply 연산자를 통해 이러한 지원이 가능함을 보였다. 그리고 구문상으로 이 연산자를 어떻게 나타낼 수 있는지에 대한 내용과 이 연산자와 기존의 관계형 연산자와의 상호작용을 제어하는 일련의 최적화 규칙들도 아울러 소개하였다.

트리-기반 스키마(tree-based schema)의 구조정보는 관계형 스키마가 가지는 테이블간의 조인에 의해 형성된다. XML 질의가 관계형 테이블에 대한 SQL 질의로 전환될 때, 간단한 XML 질의가 관계형 데이터베이스에 대한 일련의 고비용 조인들로 변환되기도 한다. 이러한 비효율적인 경우를 고려하여 XML 데이터를 관계형 테이블로 변환하지 않고, XML 데이터 관리를 직접적으로 구현한 사례들이 있다.

그 중, TIMBER 프로젝트에서는 TAX[7]라는 트리 대수(tree algebra)를 내부적으로 정의하여 잠재적으로 포함되는 그룹화 구성에 관한 내용을 지정하고, 이를 이용해 XQuery 의 내포된 질의를 TAX 의 그룹화 질의로 재구성하는 일련의 방식에 대해서 기술하였다[8]. 또한, TIMBER XML 고유 데이터베이스 시스템(TIMBER native XML database system)을 통해 이를 구현하였다.

4. XQuery 에서의 group by

본 논문에서는 기존 XQuery 에 그룹화 키워드인 **group by** 를 명시적으로 도입하는 구문 구성을 제안하고자 한다. 이를 통해 다음과 같은 효과를 얻을 수 있을 것으로 기대된다.

첫째, SQL 에서처럼 **group by** 를 사용하여 그룹화를 표현할 수 있게 된다. 현재 XQuery 에서는 내포된 FLWR 식(nested FLWR expression)과 조인을 통해 그룹화 처리를 하고 있어 질의 문장의 표현이 복잡하다.

둘째, 단일 레벨에서만 **group by** 가 가능한 SQL 과는 달리 임의의 내포 레벨에서의 그룹화가 가능해 진다. XML 의 다양한 계층 구조를 고려할 때 내포 레벨에 구애 받지 않는 그룹화가 반드시 필요하다.

셋째, 내포된 FLWR 식 안에 다시 내포된 FLWR 식을 가지는 구조를 가진 복잡한 계층 구조의 XML 문서를 쉽게 생성할 수 있게 된다.

다음에서 구체적인 예제를 사용하여 **group by** 구성을 통한 XQuery 의 확장 방안에 대해서 살펴보도록 하겠다.

그림 1 은 각 도서의 도서제목, 저자, 출판사, 가격들에 대한 정보를 담고 있는 서지목록 XML 데이터의 한 예이다.

```
<bib>
    <book year="1994">
        <title>TCP/IP Illustrated</title>
        <author><last>Stevens</last><first>W.</first></author>
        <publisher>Addison-Wesley</publisher>
        <price> 65.95</price>
    </book>
    <book year="1992">
        <title>Advanced Programming in the Unix...</title>
        <author><last>Stevens</last><first>W.</first></author>
        <publisher>Addison-Wesley</publisher>
        <price>65.95</price>
    </book>
    <book year="1994">
        <title>Data Mining <title>
        <author><last>Stevens</last><first>W.</first></author>
        <publisher>Mc Graw Hill</publisher>
        <price> 70.90</price>
    </book>
    <book year="2000">
        <title>Data on the Web</title>
        <author><last>Abiteboul</last><first>Serge</first></author>
        <author><last>Buneman</last><first>Peter</first></author>
        <author><last>Suciu</last><first>Dan</first></author>
        <publisher>Morgan Kaufmann Publishers</publisher>
        <price>39.95</price>
    </book>
    <book year="1999">
        <title>The Economics of Technology and Content...</title>
        <editor>
                <last>Gerbarg</last><first>Darcy</first>
                <affiliation>CITI</affiliation>
        </editor>
            <publisher>Kluwer Academic Publishers</publisher>
        <price>129.95</price>
    </book>
</bib>
```

그림 1. bib.xml 의 서지목록 예제 데이터

그림 2 는 XML Use Cases[9]의 1.1.9.12 Q12 를 참고하여, 그림 1 의 서지목록 예제 XML 데이터에 대해서 각 저자별, 각 년도별 출간한 도서의 제목을 검색하여 새로운 결과 XML 을 구성하는 질의를 XQuery 구문을 사용하여 작성한 것이다.

저자별, 발행년도별로 도서제목을 그룹화하기 위해 먼저 ⓐ의 for 구문에서 distinct-values 함수를 적용하여 중복된 값을 제거하고 유일한 저자노드 <author>를 추출하여 출력한다. 다시 ⓑ의 for 구문에서 distinct-values 함수를 사용하여 해당저자가 도서를 출간한 발행년도 중 중복된 값을 제거하고 유일한 발행년도만을 선별하여 <year>노드로 출력한다. 마지막으로 ⓒ의 for 구문의 where 절에서 some… satisfies… 표현식을 이용해 해당저자와 해당 발행년도를 만족하는 <book>노드들을 선별하여 각각의 노드가 가지는 도서 제목의 집합들을 새로운 <year-title>노드 안에 발행년도별로 출력한다.

그림 2 의 질의를 통해 **group by** 구성은 distinct-values 함수에 기반하고 있음을 알 수 있으며, 기존 XQuery 구문에서는 명시적인 **group by** 의 사용이 지원되지 않기 때문에 불가피하게 여러 개의 내포된 FLWR 구문과 조인을 이용할 수 밖에 없어 **group by** 를 사용한 경우보다 질의 표현이 복잡해 짐을 확인할 수 있다.

```
<results>
{
let $a := doc("bib.xml")//author
ⓐ for $last in distinct-values($a/last),
      $first in distinct-values($a[last=$last]/first)
return
   <result>
      <author>
         <last>{ $last }</last>
         <first>{ $first }</first>
      </author>
      {
   ⓑ  for $b in doc("bib.xml")/bib/book,
          $y in distinct-values($b/@year)
       where some $ba in $b/author
       satisfies ($ba/last = $last and $ba/first=$first)
       return
          <year-title>
             <year>{ $y }</year>
             {
          ⓒ  for $b1 in doc("bib.xml")/bib/book[@year=$y]
              where some $b1a in $b1/author
              satisfies ($b1a/last = $last and $b1a/first=$first)
              return $b/title
             }
          </year-title>
      }
   </result>
}
</results>
```

그림 2. **group by** 를 사용하지 않은 XQuery 질의

그림 3 은 그림 2 의 질의 문장을 **group by** 를 사용한 구문으로 구성한 것이다.

```
<results>
{
for   $b in doc("bib.xml")/bib/book,
      $a in $b/author,
      $y in $b/@year
return group by { $a, {$y} }
        <result>
           $a
           <year-title>
              <year>{ $y }</year>
                 $b/title
           </year-title>
        </result>
}
</results>
```

그림 3. **group by** 를 사용한 XQuery 질의

그림 3 에서처럼 **group by** 를 이용하는 경우에는 그림 2 에서와 같은 여러 개의 내포된 FLWR 구문과 조인을 사용하지 않고도 간결한 질의 구성이 가능하다. 특히, { } 를 사용해 $a 안에 $y 가 내포된 구조임을 표현한 것처럼, **group by** 절에 사용된 변수간의 내포관계를 정의할 수 있도록 하여, 두 개 또는 그 이상의 변수에 대해 내포된 형태의 그룹화의 표현도 손쉽게 처리되도록 한 것이 특징이다.

group by 절 구성에 지정된 바인딩 변수들에 해당하는 각각의 그룹들은 **group by** 절 구성에 사용되지 않은 바인딩 변수들에 의한 일련의 튜플들의 집합을 가진다.

그림 3 의 질의의 경우 **group by** 대상인 변수는 $a, $y 이며, **group by** 절 대상이 아닌 변수는 $b 이다. 먼저 $a 에 의한 유일한 저자의 그룹이 형성되고, 다시 $y 에 의해 해당저자 그룹 안에 유일한 발행년도 그룹이 형성된다. 마지막으로 각 해당저자의 각 발행년도 그룹에 해당하는 도서제목 튜플들이 반복적으로 생성되어 발행년도와 함께 <year-title>노드 안에 출력된다.

```
<results>
   <result>
      <author><last>Stevens</last><first>W.</first></author>
      <year-title>
            <year>1994</year>
            <title>TCP/IP Illustrated</title>
            <title>Data Mining</title>
      </year-title>
      <year-title>
            <year>1992</year>
            <title>Advanced Programming in the Unix...</title>
      </year-title>
   </result>
   <result>
      <author><last>Abiteboul</last><first>Serge</first></author>
      <year-title>
            <year>2000</year>
            <title>Data on the Web</title>
      </year-title>
   </result>
   <result>
      <author><last>Buneman</last><first>Peter</first></author>
      <year-title>
            <year>2000</year>
            <title>Data on the Web</title>
      </year-title>
   </result>
   <result>
      <author><last>Suciu</last><first>Dan</first></author>
      <year-title>
            <year>2000</year>
            <title>Data on the Web</title>
      </year-title>
   </result>
</results>
```

그림 4. 그림 1 의 예상 수행결과

5. 결론

지금까지 XQuery 및 그룹화 처리와 관련된 연구들에 대해 소개하고, XQuery 에서의 명시적인 **group by** 구성을 이용한 그룹화 처리가 필요한 이유와 **group by** 도입을 통한 XQuery 의 확장 방안을 제시함으로써, 일반적인 SQL 에서처럼 데이터의 재구성과 집계함수 처리에 유용한 그룹화를 쉽게 해주는 효율적인 질의 표현 방법에 대해 살펴보았다.

향후, 방대한 XQuery 문법 중에서 핵심적인 부분을 선별하여 **group by** 구성을 추가한 실제적인 질의 처리 시스템을 구현하고, XQuery 질의에서 **group by** 를 사용하지 않고 구성한 질의 문장을 **group by** 를 사용하여 구성한 질의 문장으로 재구성하여 처리하는 질의 최적화 규칙들을 제공하여 질의 성능의 향상시키는 방안에 대한 연구를 진행할 예정이다.

참고문헌

[1] Don Chamberlin, Jonathan Robie, Daniela Florescu, "Quilt: An XML Query Language for Heterogeneous Data Sources", WebDB, 2000.

[2] Anders Berglund, Scott Boag, Don Chamberlin, Mary F. Fernández, Michael Kay, Jonathan Robie Jérôme Siméon, "XML Path Language (XPath) 2.0", W3C Working Draft, November 2003. http://www.w3.org/TR/xpath

[3] Scott Boag, Don Chamberlin, Mary F. Fernández, Daniela Florescu, Jonathan Robie, Jérôme Siméon, "XQuery 1.0 : An XML Query Language", W3C Working Draft, November 2003. http://www.w3.org/TR/xquery

[4] D. Chatziantoniou and K. A. Ross, "Querying multiple features of groups in relational databases", VLDB(1996)

[5] D. Chatziantoniou and K. A. Ross, "Groupwise processing of relational queries", VLDB(1997)

[6] Surajit Chaudhuri, Raghav Kaushik, Jeffrey F. Naughton, "On Relational Support for XML Publishing: Beyond Sorting and Tagging", SIGMOD(2003)

[7] S.Paparizos, S. Al-Khalifa, H.V. Jagadish, L. Lakshmanan, A. Nierman, D.Srivastava, Y. Wu, "Grouping in XML", XMLDM(2002)

[8] H. V. Jagadish, L. V. S. Lakshmanan, D. Srivastava, and K. Thompson, "TAX: A Tree Algebra for XML", DBPL(2001)

[9] Don Chamberlin, Peter Fankhauser, Daniela Florescu, Jonathan Robie, Massimo Marchiori, "XML Query Use Cases", W3C Working Draft, November 2003. http://www.w3.org/TR/xquery-use-cases/