2000년 한국정보처리학회 추계 학술발표논문집 제7권 제2호

Amoeba: XML 데이터와 관계형 데이터베이스를 위한 미디에이터 시스템

박경현, 박성희, 박정석*, 류근호
충북대학교 데이터베이스 연구실
*청주 과학대학 컴퓨터과학과
{khpark, shpark, khryu}@dblab.chungbuk.ac.kr
*jspark@cjcnet.chongjunc.ac.kr

Amoeba: A Medeiator System for XML Data and Relational Databases

Kyoung Hyun Park, Sung Hee Park, Jeong Seok Park*, Keun Ho Ryu
Dept of Computer Science, Chungbuk National University
*Chongju National College of Science & Technology, Computer Science Department

요약

XML이 인터넷을 기반으로 하는 정보교환의 매개체로써 다양한 응용분야로 확산됨에 따라 XML 데이터로부터 구조정보를 추출하고 효율적으로 저장하며 관계형 데이터베이스로부터 추출된 데이터를 XML 문서로 생성하는 시스템이 요구되어진다. 기존의 관계형 데이터베이스 벤더들이 XML을 처리하기 위해 시스템을 확장하기는 하지만 이러한 시스템들은 시스템과 플랫폼에 종속적이라는 단점을 가지고 있다.

이 논문에서는 이러한 문제점을 해결함과 동시에 DTD와 관계형 스키마가 존재하지 않는 환경에서 XML문서를 효율적으로 저장하고 XML_QL을 지원하는 Amoeba 시스템을 소개한다.

1. 서론

XML이 자기 서술적(self-describing)이고 고정된 스키마가 존재하지 않는다(schemaless)는 점에서 구조적인 문서의 성격보다는 반구조적 데이터의 성격을 강하게 띄고 있다. XML의 이러한 특성은 인터넷상에서 데이터를 교환하는 매개체로서의 역할을 수행하게 하였고 그 결과 데이터 모델이나 플랫폼에 관계없이 데이터의 전송을 가능하게 하였다.

이렇듯 XML이 인터넷상에서 데이터 교환의 표준으로 자리잡으면서 기존의 관계형 데이터를 XML 형태로 전송해야 하는 필요성이 대두되었고 이에 따라 관계형 데이터베이스와 XML 문서 사이에서 데이터를 교환하는 미디에이터 시스템이 요구되었다.

미디에이터 시스템의 주요 역할은 XML 문서로부터 데이터를 추출하여 관계형 데이터베이스내에 저장하고 역으로 데이터베이스에서 추출한 데이터를 XML 문서로 생성하는 작업을 하는 것인데 이러한 작업을 하기 위해서는 DTD와 관계형 스키마가 적어도 하나는 존재해야 가능하게 된다[Bour00, Fern00, Shan99].

즉 XML 문서가 DTD를 가지고 있지 않고 관계형 데이터베이스도 관계형 스키마를 가지고 있지 않은 경우는 XML 문서를 관계형 데이터베이스내에 저장할 수 없게 된다.

이 논문에서는 이러한 문제점을 해결하기 위해 반구조적 데이터의 스키마 추출 기법과 객체-관계 매핑 기법을 이용함으로써 DTD와 관계형 스키마가 없는 환경하에서 XML 데이터를 효율적으로 저장하고 XML-QL을 지원하며 플랫폼과 데이터베이스 시스템에 독립적인 Amoeba 미디에이터 시스템을 소개한다.

2. 관련 연구

XML은 그 특성에 따라 데이터중심의 XML(data-centric XML)과 문서중심의 XML(document-centric XML)로 구분되어진다.[Bour99, Bour00].

데이터중심의 XML은 애플리케이션과 데이터 저장소간의 데이터 교환을 위해 사용되기 때문에 문서의 구조 자체가 매우 정형적이며 엘리먼트와 내용이 서로 혼합되어 있지 않다. 또한 엘리먼트들간의 순서가 중요한 의미를 가지지 않는다. 데이터중심의 XML로 제품 주문서, 비행 스케쥴등을 예로 들 수 있다.

문서중심의 XML은 계약서나 도서, 그리고 광고 같은 문서를 생성하는데 사용되기 때문에 데이터의 구조가 불규칙적이며 엘리먼트와 내용이 혼합된 경우가 많다. 또한 구조적인 문서를 표현해야 하기 때문에 엘리먼트들간의 순서가 매우 중요한 의미를 갖는다. 이러한 XML을 처리하는데 관계형 데이터베이스를 이용하는 시스템으로 XML-DBMS[Bour00]와 SilkRoute[Fern00]를 들 수 있다.

XML-DBMS는 XML을 저장하고 관계형 데이터를 XML로 변환하는 시스템으로 XML 문서를 객체 트리(object tree)로 표현되

는 뷰로 변환한 다음 객체-관계 매핑 기법[Meng95, Teor86]을 이용하여 뷰내의 객체들을 테이블로 매핑하는 방법을 사용한다. 뷰를 테이블로 매핑하고 XML 문서를 객체 뷰로 매핑하는 정보는 매핑 언어(XML-DBMS mapping language)를 통해 이루어진다.

SilkRoute는 관계형 데이터를 XML 형태로 뷰를 생성하고 질의하는 시스템으로 [Fern00]에서는 SilkRoute를 XML을 위한 일반적이고 동적이며 효율적인 시스템으로 정의하고 있다. SilkRoute는 사용자가 XML_QL로 질의를 하면 사용자 질의와 RXL(Relational to XML transformation Language)로 이루어진 뷰 질의를 바탕으로 새로운 RXL을 생성하고 이는 다시 템플릿과 데이터로그 형태로 구성되는 뷰 트리를 생성한다.

요약하면, XML-DBMS와 SilkRoute 모두 매핑언어나 RXL같은 중간적인 언어를 이용하여 XML 데이터와 관계 스키마 사이의 매핑정보를 유지하는 공통적인 특징을 가지고 있다.

그러나 XML-DBMS는 XML을 저장하기 위해서 적어도 DTD나 관계형 스키마가 존재해야 하고 SilkRoute는 관계형 데이터로부터 XML 뷰를 생성하여 질의를 처리하는 시스템이기 때문에 XML 데이터를 저장하는 모듈이 존재하지 않는다.

3. 스키마 트리(Schema Tree) 생성

관계형 스키마는 스키마 트리를 바탕으로 생성되어진다. 스키마 트리는 최대경계 스키마와 최소경계 스키마가 합성된 트리로 관계형 스키마를 생성하는데 필요한 정보를 포함한다.

3.1 최대 경계 스키마

Lore 프로젝트에서 소개된 데이터가이드는 데이터베이스 구조를 간결하고 정확하게 표현하는 스키마로 정의된다[Gold97].

즉, 데이터 소스의 모든 유일한 레이블 경로를 데이터 소스에 나타나는 빈도에 상관없이 한번만 기술하고 데이터 소스에 나타나지 않는 경로는 데이터가이드에 나타나지 않는다. 데이터가이드에 대한 이러한 특성은 반구조적 데이터의 최대경계 스키마 추출을 가능하게 해준다.

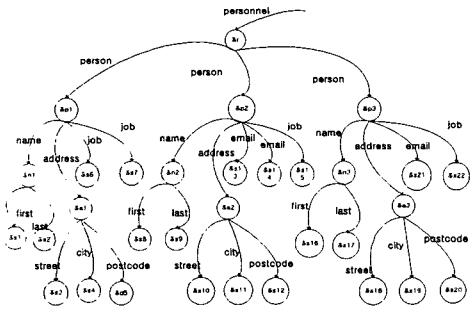

그림 1. 반구조적 데이터 모델

그림 1은 주어진 XML문서를 레이블과 방향성이 있는 반구조적 데이터 모델로 표현한 것이다. 주어진 XML 문서에 대해서 데이터가이드를 이용하여 추출한 최대경계 스키마는 그림 2와 같다.

3.2 최소 경계 스키마

시뮬레이션은 데이터 그래프와 스키마 그래프사이의 유효성을 검사하는데 적용된다. 그러나 스키마 그래프가 생성되기 이전에 주어진 데이터 그래프에 대한 스키마 그래프를 생성하기 위해서는 주어진 데이터그래프의 어떤 노드가 같은 그래프내의 다른 노드와 시뮬레이션 되는지를 판단하여 스키마 추출에 이용할 수가 있다.

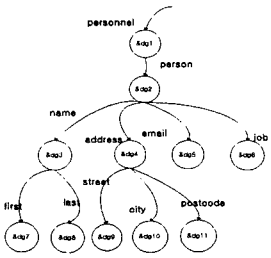

그림 2. 그림 1에 대한 데이터가이드

시뮬레이션의 이러한 성질은 주어진 데이터그래프 G에 대해 타입 정보 추출을 가능하게 한다. 즉, 하나의 그래프를 대상으로 그래프가 내포하고 있는 타입 정보를 시뮬레이션을 이용하여 추출할 수가 있다. 이러한 시뮬레이션 알고리즘은 두 단계의 과정을 통해서 주어진 데이터 그래프 G에 대한 스키마를 추출해 낸다. 시뮬레이션 알고리즘을 통해서 얻은 최소 경계 스키마는 그림 3에서 보여주고 있다.

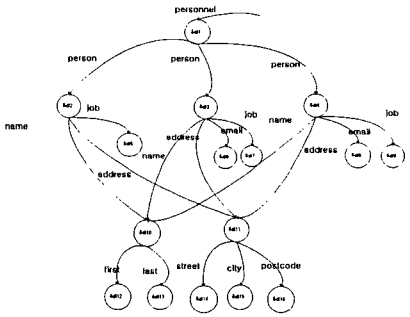

그림 3. 시뮬레이션을 이용한 최소 경계 스키마 그래프

3.3 스키마 트리

스키마 트리는 최대 경계 스키마와 최소 경계 스키마의 합성으로 생성되어진다. 그림 2의 최대 경계 스키마의 경우 스키마내에 모호성이 없는 반면 그림 3의 최소 경계 스키마는 모호성이 발생하게 된다.

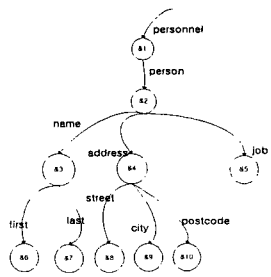

그림 4. 모호성을 없앤 최소 경계 스키마

두 개의 스키마를 합성하기 위해서는 먼저 스키마 그래프내에 모호성을 없애야 한다. 모호성을 제거한 최소 경계 스키마는 그림 4과 같다. 그림 5는 최대 경계 스키마와 최소 경계 스키마를 바탕으로 생성한 스키마 트리를 나타낸다.

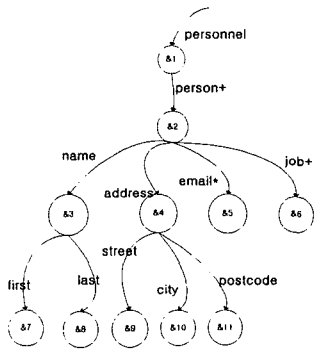

그림 5. 스키마 트리

4.관계형 스키마(Relational Schema) 생성

관계형 스키마를 생성하기 위해 객체-관계형 매핑 기법을 적용하기 위해서는 스키마 트리를 객체 트리로 인식해야 한다. 따라서 스키마 트리내의 각 노드들은 객체 트리의 관점에서 클래스(class)나 속성(property)으로 볼 수가 있다. 예를 들어, 그림 5의 &7은 속성 타입으로 간주되고 &4는 클래스 타입으로 간주되어진다. 스키마 트리의 노드가 속성 타입일때는 클래스 타입의 부모 노드에 속하게 된다. 마찬가지로 노드가 클래스 타입일때에는 스트링 값을 가진 하위 노드를 속성값으로 가진다.

이외에 관계형 스키마를 생성할 때 고려해야 할 XML 데이터의 특징으로 계층구조를 들 수 있다. 스키마 트리에서 두 노드간의 부모/자식 관계는 그 노드들의 타입에 의해 결정되어지는데 노드 타입에 따라 클래스-클래스 관계나 클래스-속성관계가 이루어진다.

표 1. &1에 대한 person 테이블

pid	first	last	street	city	postcode
p01	은선	최	사창동	청주	361-280
p02	록원	김	상동	대구	706-060
p03	은주	양	삼성동	대전	300-170

표 2. &5에 대한 email 테이블

pid	email
p01	rwkim@etri.re.kr
p01	rwkim@dblab.chungbuk.ac.kr
p02	shpark@dblab.chungbuk.ac.kr

그림 5의 스키마 트리는 객체 트리 관점에서 클래스는 테이블로 매핑되고 속성은 컬럼으로 매핑되며 클래스-클래스 관계는 후보키/외래키 관계(candidate key/foreign key relationship)로 매핑된다. 특히, 단일값 속성(single-valued property)은 클래스 테이블에서 컬럼으로 매핑되거나 BLOB로 저장하기 위해서는 별개의 테이블의 컬럼으로 매핑된다. 다중값 속성(multi-valued property)은 별개의 테이블의 다중 튜플로 매핑된다.

스키마 트리로부터 관계형 스키마를 생성시 고려해야할 또 한가지 점은 널(null) 데이터이다. XML에서 엘리먼트가 널이면 이것은 단지 문서안에 엘리먼트가 포함되지 않는 것을 의미한다. 데이터베이스에서는 이러한 엘리먼트를 처리하기 위해서 빈 문자열(empty string)을 사용해서는 안된다. 그 이유는 빈 문자열은 문자열의 길이가 0임을 나타내기 때문이다. 따라서 널 데이터를 처리하기 위해서는 컬럼을 널값을 가질 수 있도록 설정해 주어야 한다.

예를 들어, 그림 5에서 person+, job+, 혹은 email*인 레이블을 입력 간선으로 가지는 노드(&2, &5, &6)는 다중 튜플값을 가지고 있으므로 새로운 테이블을 할당하여 저장하고 만약 레이블에 ?이 존재하면 이에 더하여 매핑되는 컬럼이 널값을 가질 수 있도록 설정해야 한다. 표 1과 표 2는 그림 5의 스키마 트리를 바탕으로 생성한 관계형 스키마의 일부를 보여준다.

5. XML과 관계형 스키마간의 매핑

스키마 트리와 관계형 데이형 테이블 사이의 매핑 정보는 XML 데이터를 저장하거나 사용자가 질의를 할 때 질의에 대한 결과를 추출해내는 과정에서 요구되어진다. 이러한 매핑 정보는 XML 형태로 생성되기 때문에 일관성있는 유지가 가능해진다.

```
<class>
  <tablename elementname="person">persson</tablename>
  <column elementname="first">first</column>
  <column elementname="last">last</column>
  <column elementname="street">street</column>
  <column elementname="city">city</column>
  <column elementname="postcode">postcode</column>
  <interclass elementname="email">
    <candidatekey>pid</candidatekey>
    <foreignkey>pid</foreignkey>
  </interclass>
  <interclass elementname="job">
    <candidatekey>pid</candidatekey>
    <foreignkey>pid</foreignkey>
  </interclass>
</class>
```

그림 6. &2에 대한 매핑 정보

그림 6은 클래스 타입의 노드중 입력간선의 레이블이 person인 노드(&2)의 매핑정보를 나타낸다.

그림 6에서 보면 tablename 엘리먼트는 클래스 타입의 노드가 매핑되는 테이블명을 포함하고 tablename 엘리먼트의 elementname 애트리뷰트는 스키마 트리에서의 입력 레이블명을 나타낸다. 또한 column 엘리먼트는 테이블내의 각 컬럼에 대한 정보를 가지게 되고 interclass 엘리먼트는 스키마 트리에서 하위 노드들중에서 새로운 테이블로 생성되어지는 노드들에 대한 정보를 포함한다. interclass 엘리먼트의 하위엘리먼트인 candidatekey 엘리먼트와 foreignkey 엘리먼트는 후보키와 외래키를 각각 지정한다. 따라서 노드 &2는 xperson이라는 테이블을 생성하고 테이블내에 first, last, street등의 컬럼을 가진다. 그리고 person 테이블의 pid 컬럼은 email 테이블과 job 테이블 각각의 pid와 후보키/외래키 관계를 형성한다.

6. 질의의 최적화

사용자가 이름이 "최은선"인 사람의 이메일 주소를 검색하는

질의를 한다고 하면 질의는 아래와 같다.

```
Where <*.name>
          <first>은선</><last>최</>
          <email>$e</>
        </> in "www.a.b.c/personnel.xml"
construct <result>
            <email>$e</>
          </>
```

위의 질의를 수행하기 위해서는 first와 last 엘리먼트를 찾기 위해 모든 엘리먼트를 모두 검색해야 한다. 스키마 트리는 위와 같은 질의시에 검색의 범위를 제한함으로써 질의를 최적화시켜준다. 위의 질의에 대한 최적화된 질의은 다음과 같다.

```
Where <personnel>
        <person>
          <name>
            <first>은선</><last>최</>
          </>
          <email>$e</>
        </>
      </> in "www.a.b.c/personnel.xml"
construct <result>
          <email>$e</>
        </>
```

7. 시스템 구조

그림 8은 Amoeba 시스템의 구조를 보여준다. 우선 XML 문서가 들어오면 스키마 추출기(Schema Extractor)는 스키마 트리를 생성하고 테이블 생성기(Table Generator)는 스키마 트리를 입력받아 SQL을 생성하여 데이터베이스에 테이블을 생성한다. 테이블이 생성되면 XMLtoDB는 매핑정보를 이용하여 데이터베이스에 XML 데이터를 저장한다.

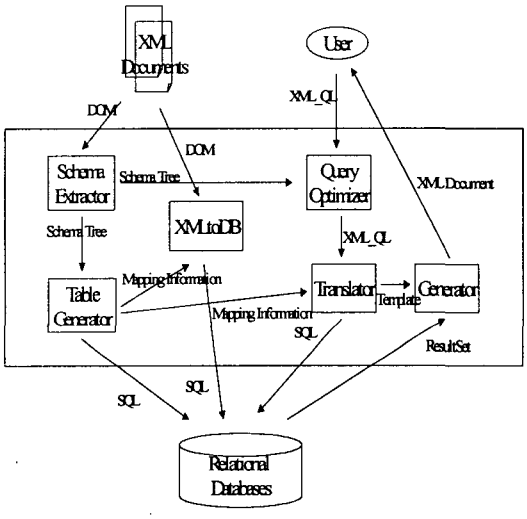

그림 8. Amoeba 시스템 구조

사용자는 스키마 트리로 생성한 XML 뷰를 대상으로 질의를 하게 되고 질의는 질의 최적기(Query Optimizer)에 의해 최적화되어진다. 최적화된 질의는 번역기(Translator)에서 다시 where절과 construct절로 분해되어 2개의 트리로 형성된다. 그 중 where절에서 추출한 트리는 스키마 트리와 매핑 정보를 이용하여 SQL을 생성하고 관계형 데이터베이스에 질의를 한다. 질의 결과는 생성기(Generator)를 통해 construct절에서 추출한 템플릿 트리에 매핑되어 XML뷰로 변환되고 사용자에게 전달되어진다.

8. 결론 및 향후 연구방향

이 논문에서는 DTD와 관계형 스키마가 존재하지 않는 환경에서 XML 데이터를 효율적으로 저장하기 위한 미디에이터 시스템을 설계하였다. Amoeba 시스템은 데이터 중심의 XML 성격이 강한 유전체 데이터베이스의 XML 지원을 위해 설계되었기 때문에 현재 문서 중심의 XML의 특징인 엘리먼트들간의 순서 및 링크, 속성(attribute)과 하위 엘리먼트(subelement)의 구분등이 고려되지 않았다.

따라서 향후 본 시스템의 구현 및 성능 평가를 통한 추가적인 연구이외에도 데이터 중심의 XML에 한정되지 않은 범용적인 미디에이터 시스템으로 확장하기 위한 연구가 필요하다.

참고문헌

[Bour99] Ronald Bourret. XML and Databases. Technical University of Darmstadt. "http://www.informatik.tu-darmstadt.de/DVS1/staff/bourret/xml/XMLAndDatabases", 1999

[Bour00] R. Bourret, C. Bornhovd, A. P. Buchmann. A Generic Load/Extract Utility for Data Transfer between XML Documents and Relational Databases, WECWIS'00, San Jose, California, June 8-9, 2000

[Fern00] Mary Fernandez WangChiew Tan Dan Suciu, SilkRoute: Trading between Relations and XML, WWWg, 2000

[Gold97] R. Goldman, J. Widom. DataGuide : Enabling Query Formulation and Optimization In Semistructured Databases. In Proc. of the 23rd VLDB Conference Athens, Greece, 1997

[Meng95] W. Meng, A. Kamada, Y-H Chang, Transformation of Relational Schemas to Object-Oriented Schemas, COMPSAC, Dallas, Texas, 1995

[Shan99] Jayavel Shanmugasundaram, Kristin Tufte, Gang He, Chun Zhang, David DeWitt, Jeffrey Naughton, Relational Databases for Querying XML Documents:Limitations and opportunities, In Proc. of VLDB Edinburgh, Scotland, 1999.

[Teor86] T. Teorey, D. Yang, J. Fry: A Logical Design Methodology for Relational Databases Using the Extended Entity-Relationship Model, ACM Computing Surveys, Vol.18, No.2, 1986

[박성희00] 박성희, 박경현, 김록원, 남광우. ORDBMS를 이용한 XML 문서 저장 및 질의. 제27회 한국정보과학회 춘계학술대회, 2000

[양은주00] 양은주, 박경현, 류근호. XML 데이터의 효율적인 DTD 추출. 제14회 한국정보처리학회 추계학술대회, 2000