멀티데이터베이스에서 XML을 이용한 스키마 통합

탁 우현, 류 영호, 이 종환, 김 경석

부산대학교 전자계산학과

Schema Integration Using XML in Multi-Database

Tak Woohyeon, Ryu Youngho, Lee Jonghwan, Kim Kyongsok

Dept. of Computer Science Pusan National Univ.

요약

멀티데이터베이스를 구축하기 위해서는 전역스키마 (global schema) 구축이 필수적이며, 지역스키마 (local schema) 간 충돌 해결과 스키마 통합 정보는 질의 처리 및 질의 결과를 사용자에게 보여주기 위해 필요하다. 전역스키마 생성 및 충돌 해결 방법은 많이 연구가 되어 왔다. 하지만, 스키마 통합을 위한 정보가 어떻게 표현되고, 어떻게 사용되는지에 대한 연구는 부족하다. 본 논문에서는 지역스키마 와 전역스키마를 XML DTD 및 XML 문서로 표현하고, 이를 이용하여 멀티데이터베이스를 위한 스키마 통합 방법을 제안한다. 스키마 통합 정보와 멀티데이터베이스와 지역데이터베이스들간 데이터 교환을 XML을 이용함으로써 이기종간 데이터 교환이 쉽고 멀티데이터베이스 시스템의 확장 및 다른 시스템으로의 전환이 용이하게 된다.

1. 서론

XML (eXtensible Markup Language) 은 사용자가 원하는 정보를 검색하는 다양한 방법을 제공할 뿐만 아니라, 웹에서 사용할 수 있는 메타데이터를 구조화하여 많은 웹 기반 응용 프로그램에 사용될 수 있다.

XML은 정보 사용자들과 제공자들이 상대방을 쉽게 찾을 수 있고, 검색이나 정보 교환과 같은 작업이 자동화되고, 정보를 표현하는 일반적인 프레임워크를 제공한다. 그리고, 한가지 형태로 된 컨텐트를 마크업하는데 유용한 언어이다. 또한, XML은 모든 플랫폼, 운영체제 환경에서 실행할 수 있는 마크업 언어로서 웹의 컨텐트를 더 효율적으로 표현하도록 고안된 것이라고 할 수 있다[1]. 이러한 특징으로 말미암아 데이터를 좀 더 쉽고 효과적으로 교환할 수 있는 수단으로 발전하였다.

초기 데이터베이스 시스템은 단일 운영체제 및 시스템에서 데이터를 액세스 했다. 하지만, 데이터베이스 시스템의 종류가 다양해지고 시스템의 개선 및 확장으로 서로 다른 데이터베이스 시스템으로부터 데이터를 액세스 해야할 필요가 생겼다[5].

멀티데이터베이스는 이기종 데이터베이스의 데이터를 단일 전역스키마를 통해 액세스 한다. 이때 가장 중요한 것은 서로 다른 지역 데이터베이스에서 같은 데이터에 대해 서로 다르게 표현된 것을 전역 데이터베이스에서는 같은 데이터로 인식하게 하는 것이다.

본 논문에서는 멀티데이터베이스에서 전역스키마를 XML 문서로 표현하여 지역스키마 통합과 충돌 해결방법에 대한 정보를 전역스키마인 XML 문서에 포함시키는 방법을 제안한다. 스키마 통합과 데이터 교환을 위해 XML을 이용함으로써 이기종[1] 시스템간 데이터 교환이 쉽고, XML 문서 자체에 구조를 포함함으로써 지역스키마 구조를 파악하기 쉽다. 그리고, 전역스키마에서는 충돌해결 및 통합방법에 대한 정보를 XML 문서 내에 포함시킴으로써 질의 처리시 전역 뷰 (global view) 를 통해 바로 질의 분할 (query decomposition) 이 가능하다.

2장에서 관련연구를 살펴보고, 3장, 4장에서 전역스키마에 대한 DTD와 충돌해결방법을 설명하고, 5장에서 결론 및 향후 연구과제를 설명하겠다.

2. 관련 연구

2.1 XML

XML은 마크업을 하기 위한 규칙을 규정하는 DTD (Document Type Definition) 와 태그 (tag) 가 붙은 XML 문서로 나눌 수 있다. 유효한 문서는 DTD가 정의한 구조와 규칙을 철저히 준수해야 한다[2]. DTD는 문서의 구조적 정보를 표현하며, XML 문서는 DTD를 기반으로작성함으로써 구조화된 문서로 표현된다. 이러한 특징때문에 동일한 구조를 가지는 컨텐트를 표현하는 것에 많이 사용된다. 본 논문에서는 지역스키마 와 전역스키마를 XML 문서로 표현하여 스키마 구조를 쉽게 파악하고, 시스템간 데이터 교환이 쉽도록 한다.

2.2 멀티데이터베이스

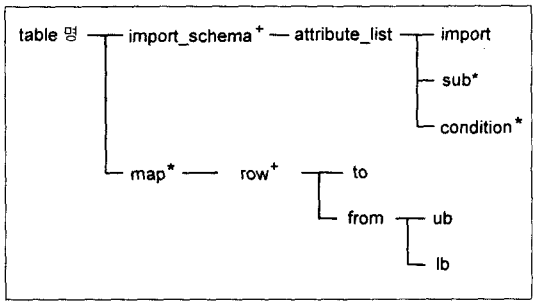
1) RDB, ORDB, OODB를 모두 포함한다.

멀티데이터베이스는 지역 데이터베이스로부터 스키마 정보를 전달받아 전역스키마를 생성한다. 같은 개념스키마에 대한 지역 데이터베이스에서의 다른 표현으로 생기는 스키마 충돌과 해결방법은 오래 전부터 연구되어 왔고 현재도 새로운 해결방법이 제안되고 있다. [6]은 개체-관계 (entity-relationship) 모델을 통합하는 방법에 대해 설명하고, [7]은 통합을 위한 정보를 표현하기 위해 또 다른 데이터베이스에 구조화하여 저장한다. [8]은 서로 다른 지역스키마 모델을 단일 스키마 모델로 변환한 다음 통합하는 단계를 거친다. 이와 같이, 기존 연구들은 통합하는 방법에 대한 연구는 많았다. 하지만, 통합과 충돌해결을 위한 정보가 실제 시스템 구현시 어떻게 표현되고, 어떻게 사용되는지에 대한 연구는 부족하다. 본 논문에서는 통합시 필요한 정보나 구조를 표현하고 저장하기 위해 XML을 이용하는 방법을 제안한다.

3. 전역스키마와 DTD

멀티데이터베이스 시스템은 지역 데이터베이스 시스템으로부터 전달 (import) 받은 지역스키마를 통합하여 하나의 전역스키마를 생성한다. 지역스키마는 XML 문서화되어 전송되고 시스템 관리자는 이 문서를 전역스키마를 표현하는 XML 문서로 만들게 된다. 전역스키마에 대한 XML 문서는 스키마 구조뿐만 아니라, 지역스키마 통합 방법과 충돌 해결 방법에 대한 정보까지 가지게 된다. 스키마 통합을 위한 정보를 전역스키마 구조에 포함함으로써 지역스키마에 대한 정보를 알 수 있으며, 질의 처리시 뷰를 참조함과 동시에 질의를 분할 할 수 있다.

[그림 1] 에서는 전역스키마에서 하나의 테이블에 대한 XML DTD 구조를 보여준다.



[그림 1] 전역스키마의 테이블 하나에 대한 DTD 구조

① *table명*

전역스키마에서 해당 테이블명을 가리킴. 실제 해당 테이블 DTD에서는 테이블명이 요소명이 된다. 전역스키마의 모든 테이블은 [그림 1] 과 같은 DTD를 만족하는 XML 문서를 가진다.

② *import_schema*

지역스키마로부터 가져올 스키마를 표현하는 요소. 질의 결과에 대한 데이터를 표현할 때는 지역스키마에서 가져온 하나의 튜플을 가리키는 요소.

③ *attribute_list*

테이블의 일련의 애트리뷰트명이 요소명이 된다. 실제 해당 테이블의 DTD에서는 <attribute_list>는 해당 애트리뷰트들의 이름으로 대치된다.

④ *import*

지역스키마로부터 통합하는 규칙 및 충돌 해결방법을 설명하는 요소.

⑤ *sub*

import 요소에 추가하여 통합규칙을 정의하는 요소. 다수의 스키마 충돌이 생겼을 때 사용하게 된다.

⑥ *condition*

지역스키마로부터 애트리뷰트를 액세스하기 위한 조건절 표현하는 요소.

⑦ *map*

충돌해결방법 중 변환표 (mapping table) 가 필요한 경우 변환표를 설명하는 요소.

[그림 1]의 DTD 구조에는 *'table 명', 'attribute_list'* 이라는 요소명이 있지만 실제 DTD는 해당 테이블명을 요소명으로 갖고, 하나의 테이블에 존재하는 모든 애트리뷰트가 요소명으로 DTD 내에 들어간다.

4. XML을 이용한 충돌해결

데이터베이스의 용도와 설계자가 다름에 따라 같은 데이터 모델에 대해 서로 다른 스키마를 가질 수 있다. 이러한 이유로 지역스키마 통합시 스키마 충돌 해결이 선행되어야 한다. 스키마 통합시 발생할 수 있는 충돌과 이에 대한 해결방법은 [4]에서 정의한 방법을 사용한다. 충돌해결방법에 대한 정보는 전역스키마 DTD의 *import, sub, condition, map* 요소를 사용하여 표현한다.

[그림 2] 는 스키마 DTD에 대한 일부분인 *import* 요소에 대한 명세를 나타내고 있다.

```
<!ELEMENT import  EMPTY >
<!ATTLIST import
        id          ID          #REQUIRED
        sub_refs    IDREFS      #IMPLIED
        con_refs    IDREFS      #IMPLIED
        table       CDATA       #REQUIRED
        name        CDATA       #REQUIRED
        type        CDATA       #REQUIRED
        length      CDATA       #REQUIRED
        category    (DE|DP|DU|CA|
                    CC|NA|MI|NN) "NN"
        operator    (M|D|S|A)   #IMPLIED
        operand     CDATA       #IMPLIED
        reference   CDATA       #IMPLIED" >
```

[그림 2] import 요소에 대한 DTD 명세

속성 *id , sub_refs, con_refs*는 *sub, condition*과의 양방향 링크를 위해 사용된다. *table*과 *name* 속성은 지역스키마의 테이블명과 애트리뷰트명을 가리킨다. *type*과 *length*는 지역스키마의 데이터형 (data type) 을 가리키며, *category* 속성은 어떤 충돌인지를 가리키며 그 값에 따라 *operator, operand, reference*의 속성값을 이용하여 충돌을 해결한다.

① 이름 (naming) , 데이터형 충돌

테이블이나 애트리뷰트명이 의미적으로는 같지만 다른 이름이 사용된 경우와 애트리뷰트의 데이터형이 다를 경우.

```
<name id="a001" type="char" length="10" >
   <import id="i01" table="stud" name="sname"
        type="char" length="5" />
</name>
```

지역 데이터베이스의 테이블 (stud) 은 char(5) 데이터형을 가진 애트리뷰트 (sname) 를 가졌지만 전역스키마에

서는 char(10) 데이터형의 애트리뷰트 (name) 로 변환됨을 의미한다.

② 같은 데이터에 대한 서로 다른 표현

같은 데이터를 서로 다르게 표현함으로써 생기는 충돌. 이 같은 경우는 변환표가 필요하게 되는데 *map* 요소를 이용하게 된다.

```
<grade id="a001" type="char" length="15" >
   <import id="i01"  table="stud" name="grade"
        category="DE" reference="m01" />
</grade>

<map id="m01">
   <row id="r01">
        <to> EXCELLENT </to>
        <from> A </from>
   </row>
   <row id="r02">
        <to> GOOD </to>
        <from> B </from>
   </row>
   .. 중간 생략 ..
</map>
```

지역스키마 테이블 (*stud*) 은 학생의 등급을 'A', 'B' 로 했으나 전역스키마는 'EXCELLENT', 'GOOD'으로 한 경우이다. 만약 *stud.grade* = 'A'이면, 'A' 값을 가지는 from 요소와 쌍을 이루는 to 요소값 'EXCELLENT'로 변환된다.

③ 서로 다른 단위

같은 데이터에 대해 서로 다른 단위를 사용함으로써 생기는 충돌.

```
<weight id="a001" type="integer"  length="3">
   <import  id="i01"  table="stud" name="weight"
     category="DU" operator="M" operand="2.25" />
</weight>
```

지역스키마 테이블 (stud) 는 몸무게 (weight) 를 kg 단위를 사용하였지만 전역스키마는 파운드 (pound) 를 사용한 경우 weight * 2.25 의 값으로 통합된다. operator는 사칙연산을 표현하며 G = a * b + c 연산이 필요한 경우 *import* 요소를 사용하여 A = a * b 를 표현하고 추가적인 연산은 *sub* 요소를 사용하여 G = A + c 를 표현한다.

④ 합침 (concatenation)

구조적 충돌중의 하나이다. 전역스키마에서는 하나의 애트리뷰트로 정의되었지만 지역스키마에서는 한 개 이상의 애트리뷰트로 정의되었을 때 생긴다.

```
<name id="a001" type="char" length="10" />
  <import id="i01" sub_refs="s01" table="stud"
  name="fname" category="CC" reference="middle" />
  <sub id="s01" category="CC" reference="lname" />
</name>
```

지역스키마의 테이블 (stud) 은 학생의 이름을 'firstname, middle, lastname' 으로 나누어 정의했지만 전역스키마는 단일 애트리뷰트 (name) 로 정의한 경우이다.

[표 1] 은 [5]에서 정의한 충돌과 해결방법에 따른 *category* 요소값과 관련 속성을 보여준다.

DE, DP 요소는 *reference* 요소의 변환표를 이용하고, DU 요소는 *operator, operand* 요소를 이용하여 서로 다른 단위 (data unit) 를 가지는 경우 연산에 의해 단위를 통일한다. CC, CA 요소는 구조적 충돌로 인하여 애트리뷰트를 합치거나 나누어야 할 경우이다. MI, NA 는 애트리뷰트는 없지만 내포하는 의미가 있을 경우와 그렇지 않은 경우이며, 위의 경우에 속하지 않으면 기본값인 NN을 가진다.

[표 1] 충돌 종류에 따른 category 요소값

category	관련 속성	충돌 종류
DE	reference	같은 값에 대한 다른 표현
DP	reference	값에 대한 다른 정확도 (precision)
DU	operator, operand	값에 대한 다른 단위
CA	reference	복합 (composite) 애트리뷰트
CC	reference	합쳐야 (concatenation) 될 애트리뷰트
MI	reference	애트리뷰트는 없지만 뜻을 내포한 경우
NA	없음	애트리뷰트가 없어 불가능한 경우
NN	type, length	서로 다른 데이터형
NN	name	이름 충돌

5. 결론 및 향후 연구

본 논문에서는 멀티데이터베이스 구축을 위한 전역스키마를 XML DTD로 정의하였다. 스키마 통합과 충돌 해결방법을 전역스키마인 XML 문서내에 표현함으로써 질의 처리시 전역 뷰를 참조함과 동시에 질의 분할이 가능하다. 구조화된 XML 문서를 통해 전역스키마와 지역스키마간의 관계를 쉽게 파악할 수 있다. 스키마 정보 및 데이터를 정형화된 XML 문서 형식으로 교환함으로 정보에 대한 분석이 쉽고 이기종간 데이터 교환이 용이하다. 또한, 질의 결과에 대한 사용자 인터페이스도 결과 데이터가 XML 문서이므로 XSL을 이용한 다양한 프리젠테이션이 가능하게 된다.

향후 연구과제로는 지역 데이터베이스에서 전달 (export) 해야하는 스키마 구조를 DTD로 정의해야 하며, 전역스키마의 제약 조건 (constraints) 을 표현할 수 있는 방법을 연구해야 한다. 그리고, 전역 뷰를 통한 사용자 인터페이스를 개발해야 한다.

6. 참고 문헌

[1] W3C XML Activity
http://www.w3.org/XML/Activity.html

[2] W3C XML Spec.
http://www.w3.org/TR/1998/REC-xml-19980210.html

[3] W3C XML Stylesheet Language
http://www.w3.org/Style/XSL/

[4] Jon Bosak, XML, Java, and the future of the Web
http://metalab.unc.edu/pub/sun-info/standards/xml/why/xmlapps.htm

[5] Won Kim, Modern Database Systems: The Object Model, Interoperability and Beyond, ACM Press,1995

[6] LEE, M.L. and Ling, T.W., Resolving Structural Conflicts in the Integration of Entity-Relationship Schemas, Proc. of the 14th Int. Conf. on OOER'95, Gold Coast, Australia, pp. 422-433.

[7] J. Yang, Mike P. Papazoglou, A Configurable Approach for Object Sharing among Multidatabase Systems, CIKM'95, Baltimore MD USA pp. 129-136

[8] Pyeong S. Mah, Soon M. Chung, Schema integration and transaction management for multidatabases, Information Sciences 111(1998) pp.153-188

[9] I. Schmitt., S. Conrad, Restructuring Class Hierarchies for Schema Integration, Proc. of the 15th Int. Conf. on DSAA, Australia, pp. 411-420