

인터넷 데이터웨어하우스 구축을 위한 메타데이터 스키마 충돌 제어

김병곤*

요약

인터넷을 통한 웹서비스에 대한 사용자들의 요구가 증가하고 다양해지면서 정보들을 수집하고 통합하여 특정집단 또는 일반인들의 의사결정을 지원하기 위한 인터넷 데이터웨어하우스에 대한 중요성이 갈수록 증대되고 있다. 초기의 인터넷 데이터웨어하우스는 기존의 데이터베이스와 XML을 이용한 형태로 연구되었으나, 정보 표현상의 한계로 인하여 점차 RDFS와 같은 메타데이터 스키마를 이용한 시스템 형태로 변화하고 있다. 인터넷의 특성상 분산된 시스템 환경에서 서로 상이한 메타데이터 스키마들을 통합과 저장이 중요한 요소이다. 그러나, 서로 상이한 스키마간에는 충돌할 수 있는 요소들이 존재하므로 지역 스키마들 간에 발생할 수 있는 의미적, 구조적 충돌을 감지하고 고려하여 저장이나 질의 처리 등에서 완벽한 처리가 가능하도록 시스템을 구성하여야 한다. 본 논문에서는 이와 같이 분산된 환경에서 지역 메타데이터 스키마들 간의 통합시 충돌이 발생하는 경우를 분석하고, 이에 따른 충돌 해결 방법을 제시하여 완벽한 스키마 통합이 가능하도록 한다.

Control of metadata schema conflicts for internet datawarehouse

Byung-gon Kim*

Abstract

With the increasing of users' request about internet web service, importance of Internet datawarehouse to support decision making of users is increasing now. Early Internet datawarehouse was studied in the form of using existent database and XML. However, because of limitation of information expression ability, it is gradually changed to system that use metadata schema like RDFS. Because of distributed environment of the Internet, integration and saving of each metadata schemas into one global schema is important. However, between each different schema, semantic and structural conflicts can be happen in such situation and they must be controlled. In this paper, we analyze occasions of conflict when integrate distributed metadata schemas and propose conflict resolution technique for efficient internet datawarehouse query processing.

Keywords : Internet Datawarehouse, XML, RDF, RDF/S, Schema conflict

1. 서론

전 세계적으로 인터넷의 사용이 일반화 되면서 인터넷상의 정보들을 수집하고 통합하여 특정집단 또는 일반인들의 의사결정을 지원하기 위한 인터넷 데이터웨어하우스에 대한 중요성이

갈수록 증대되고 있으며 이에 대한 연구가 활발히 진행되고 있다. 초기의 인터넷 데이터웨어하우스는 데이터의 검색, 전송 및 저장을 위하여 XML의 구조적인 장점을 이용하여 데이터웨어하우스를 구성하고 이를 통하여 효율적인 인터넷 데이터 통합시스템을 구축하였다[1][2][3][4]. 그러나, XML 문서만으로 인터넷 데이터웨어하우스 시스템을 구축하는 데는 정보간의 특성이나 관계 등을 표현하는데 한계가 있다. 따라서, 점차적으로 RDF, DAML+OIL, OWL[5][6][7]등의 메타데이터 스키마 언어를 이용하여 XML 데이터웨어하우스의 한계를 극복하고, 효율적인 검색이 가능한 시맨틱 웹 환경의 데이터웨어하우스 구축에 대한 연구로 진행되고 있다.

※ 제일저자(First Author) : 김병곤
접수일자:2007년10월22일, 심사완료:2007년10월25일
* 부천대학 e-비즈니스과
bgkim@bc.ac.kr
■ 이 논문은 2005년도 정부재원(교육인적자원부 학술연구조성사업비)으로 한국 학술진흥재단의 지원을 받아 연구되었음(KRF-2005-003-D00314)

시맨틱웹 환경의 데이터웨어하우스 구축에 있어서 관심분야중 하나는 각각 여러 사이트에서 개발되고 사용되어지고 있는 데이터웨어하우스의 데이터와 데이터에 대한 스키마를 하나의 환경으로 통합하여 질의하고 검색할 수 있는 시스템을 구축하는 것이다. 이러한 분산된 데이터의 스키마와 데이터 통합 문제는 차세대 인터넷 기술의 핵심 요소가 될 것이며, 신뢰성 있는 시스템 구축을 위한 중요한 역할을 할 수 있다.

여러 곳에 분산되어 있는 메타데이터들을 통합하여 전역질의가 가능하게 하려면, 분산된 환경에서 각 사이트마다 구성되어 있는 지역스키마를 통합 관리할 수 있는 전역 스키마의 구축이 중요한 역할을 담당한다. 전역 스키마를 구축할 때 중요한 요소들은 스키마 통합에 따른 저장구조, 인덱스구조, 질의처리 기법 등이 고려되어야 하며, 지역 스키마들 간에 발생할 수 있는 의미적, 구조적 충돌을 감지하여 저장이나 질의처리 등에서 완벽한 처리가 가능하도록 시스템을 구성하여야 한다는 것이다. 스키마의 충돌은 단순히 클래스나 프로퍼티의 이름이 충돌하는 경우에서부터 클래스의 상하관계나 도메인 등이 상이한 경우까지 다양하다.

데이터베이스나 XML 스키마 환경에서의 충돌해결 기법에 대한 연구는 진행되었으나, RDF 스키마에 대한 통합시 충돌 해결 기법은 구체적으로 연구되지 못하였다. 본 논문에서는 분산된 환경에서 지역 메타데이터 스키마들 간의 통합시 충돌이 발생하는 경우를 분석하고, 이에 따른 충돌 해결 기법을 제시하여 완벽한 스키마 통합이 가능하도록 한다. 또한 제안된 충돌 해결기법을 구현하기 위한 RDB의 테이블 형태의 정보 저장 구조를 제시하였다.

2. 본론

2.1 메타데이터 스키마통합 관련연구

웹상의 데이터웨어하우스의 구축을 위한 메타데이터 통합시스템의 초기에는 기존의 데이터베이스 시스템을 기초로 구성된 온톨로지를 구성요소로 하는 환경에서 스키마를 이용한 구축과 통합의 연구가 주를 이루었다. SCROL 시스템[8]은 온톨로지 스키마간의 의미적 충돌을 감지

하고 해결하기 위한 방법을 제시하였다. 먼저, 온톨로지의 의미적인 충돌을 데이터 레벨에서의 충돌과 스키마 레벨에서의 충돌로 분류하였다. 특히 스키마 레벨의 충돌을 이름충돌, 엔티티 식별자 충돌, 스키마 동형 충돌(Schema isomorphism conflict), 일반화 충돌(Generalization conflict), 어그리게이션 충돌(Aggregation conflict), 등으로 분류하고, 이에 대한 해결 방안을 제시하였다. 그러나, SCROL 시스템은 각각의 온톨로지 소스 데이터가 기본적으로 데이터베이스 시스템으로 구성되어 있는 경우에 적용하여 진행되었다. 따라서, 이러한 이론들을 웹상의 시맨틱웹 데이터인 XML이나 RDF로의 확장이 요구된다.

[9]에서는 미디어데이터 환경에서 XML 스키마를 이용하여 통합 스키마를 얻기 위한 시스템을 구성하는데 있어서 스키마의 충돌을 이름충돌, 타입충돌, 제약충돌로 분류하였다. 이름충돌(Naming Conflict)은 같은 개념을 다른 용어를 사용하여 표현한 경우와, 혹은 다른 개념을 같은 용어를 사용하여 표현한 경우이다. 타입충돌(Type Conflict)은 의미적으로 같은 엘리먼트나 애트리뷰트가 서로 다른 구조나 데이터타입을 가지는 경우에 발생한다. 제약충돌(Constraints Conflict)은 의미적으로 같은 엘리먼트나 애트리뷰트가 서로 다른 제약사항을 가지는 경우이다. 이름 충돌은 중앙의 카탈로그에서 이름을 재명명하여 해결하는 방법을 사용하였고, 타입 충돌이 발생하면 엘리먼트의 옵션의 선택이 가능하도록 하거나, 데이터 타입의 자동변환을 이용하는 방법을 제시하였다.

현재의 데이터웨어하우스의 통합 시스템의 통합스키마는 RDF와 같은 메타데이터 전용 언어를 중심으로 연구되어지고 있다. RDF와 같은 온톨로지 언어를 근간으로 하는 데이터웨어하우스의 통합의 연구는 기구축된 데이터웨어하우스 내의 온톨로지를 대상으로 하여 새로운 통합된 온톨로지를 구축하기 위하여 온톨로지간 레벨에 따른 비교 연구[10][11]와 온톨로지간 통합 방법론에 대한 연구[12][13]로 진행되고 있다.

그러나, RDF나 OWL과 같은 메타데이터 언어로 표현된 스키마의 충돌은 단순한 XML의 충돌과 달리 각 요소간의 관계의 충돌도 고려하여 제어하여야 하며, 통합 측면의 연구에서 더

나아가 적극적인 충돌 관리 메커니즘을 필요로 한다. 즉, 클래스, 프로퍼티, 도메인, 레인지 등의 세부적인 요소들에 대한 충돌 해결 방안이 요구되며, 이러한 충돌의 해결 없이는 전역 스키마에 대한 질의를 수행할 수 없다. 본 연구에서는 먼저 RDF 스키마의 충돌의 종류를 분류하고 이에 따른 해결책을 제시하였다.

2.2 RDF 스키마 충돌 분류

본 연구에서는 RDF 스키마를 대상으로 한 스키마 통합 시 발생할 수 있는 충돌의 종류를 이름충돌, 관계충돌, 타입충돌로 분류하였으며, 각각의 의미를 살펴보면 다음과 같다.

▶ 이름 충돌(Naming Conflict)

- 클래스 이름(Class name)
 - 이음동의클래스 : 동일한 의미의 클래스가 다른 스키마에 다른 이름으로 존재하는 경우
 - 동음이의클래스 : 다른 의미의 클래스가 다른 스키마에 같은 이름으로 존재하는 경우
- 프로퍼티 이름(Property name)
 - 이음동의프로퍼티 : 동일한 의미의 프로퍼티가 다른 스키마에 다른 이름으로 존재하는 경우
 - 동음이의프로퍼티 : 다른 의미의 프로퍼티가 다른 스키마에 같은 이름으로 존재하는 경우

▶ 관계 충돌(relation conflict)

- 서브클래스충돌(SubclassOf conflict) : 동일한 클래스의 서브클래스 요소가 상이한 경우. 예를 들어 동일한 클래스 Taxi의 서브클래스요소가 스키마에 따라 아래와 같이 다를 수 있다.

```

Schema A
<rdf:Description ID="Taxi">
<rdf:type resource=
"http://www.w3.org/2000/01/rdf-schema#Class"/>
<rdfs:subClassOf rdf:resource="#MotorVehicle"/>
</rdf:Description>
Schema B
<rdf:Description ID="Taxi">
<rdf:type resource=
"http://www.w3.org/2000/01/rdf-schema#Class"/>
<rdfs:subClassOf rdf:resource="#Vehicle"/>
    
```

</rdf:Description>

- 서브프로퍼티충돌(SubpropertyOf conflict) : 동일한 프로퍼티의 서브프로퍼티 요소가 상이한 경우. 예를 들어 동일한 프로퍼티 biologicalFather의 서브프로퍼티요소가 스키마에 따라 아래와 같이 다를 수 있다.

```

Schema A
<rdf:Description ID="biologicalFather">
<rdf:type resource=
"http://www.w3.org/1999/02/22-rdf-syntax-ns
#Property"/>
<rdfs:subPropertyOf
rdf:resource="#biologicalParent"/>
</rdf:Description>
Schema B
<rdf:Description ID="biologicalFather">
<rdf:type resource=
"http://www.w3.org/1999/02/22-rdf-syntax-ns
#Property"/>
<rdfs:subPropertyOf
rdf:resource="#biologicalAncestor"/>
</rdf:Description>
    
```

- 도메인/레인지 충돌(Domain/Range conflict) : 동일한 프로퍼티의 도메인이나 레인지 요소가 상이한 경우. 아래의 예에서는 age라는 이름의 프로퍼티에 대한 도메인이 스키마에 따라 Person 과 Animal로 서로 상이하어 충돌이 발생하였다.

```

Schema A
<rdf:Property ID="age">
<rdfs:range rdf:resource=
"http://www.w3.org/2000/03/example/classes
#Integer"/>
<rdfs:domain rdf:resource="#Person"/>
</rdf:Property>
Schema B
<rdf:Property ID="age">
<rdfs:range rdf:resource=
"http://www.w3.org/2000/03/example/classes
#Integer"/>
<rdfs:domain rdf:resource="#Animal"/>
</rdf:Property>
    
```

- ▶ 타입 충돌 : 동일한 단어가 스키마에 따라

타입이 다를 때이다. Register라는 이름이 각 사이트의 스키마에 따라 클래스 이름으로 사용될 수도 있고, 프로퍼티의 이름으로 사용될 수도 있다.

```

Schema A
<rdf:Description ID="Register">
<rdf:type resource=
"http://www.w3.org/2000/01/rdf-schema#Class"/>
<rdfs:subClassOf rdf:resource=
"#ComputerComponent"/>
</rdf:Description>
Schema B
<rdf:Description ID="register">
<rdf:type resource=
"http://www.w3.org/1999/02/22-rdf-syntax-ns
#Property"/>
<rdfs:domain rdf:resource="#Person"/>
<rdfs:range rdf:resource="#Product"/>
</rdf:Description>
    
```

위에서 분류한 충돌들은 분산된 환경의 스키마에서 항상 발생할 수 있는 사항이며 통합된 중앙의 스키마를 유지하여 온톨로지를 구축하려면 각각의 충돌에 대한 대비책을 마련하고 미리 질의에 대응하도록 시스템을 구성하여야 한다.

2.3 스키마 충돌 해결

분산된 환경에서 스키마간에 발생할 수 있는 충돌을 어떤 형태로든 해결하고, 질의 처리가 가능한 환경을 구성하기 위하여, 이 장에서는 앞장에서 분류한 충돌의 종류별로 해결방법을 제시한다.

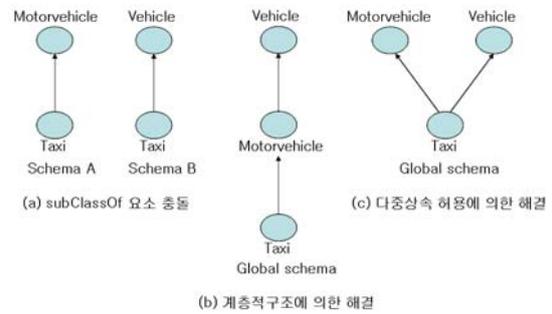
▶ 이름 충돌(Naming Conflict)의 해결

분산된 사이트의 스키마간에 이름충돌이 발생하면, 각 지역스키마의 이름을 파악하고 매칭시킬수 있는 카탈로그 형태의 정보를 유지한다. 이때 각 사이트의 스키마내의 이름들을 구분할 수 있도록 충돌이 발생한 이름들을 재명명(Renaming)하여 충돌을 제어한다. 카탈로그는 광역스키마(Global Schema)에 대한 질의를 처리하는 경우 반드시 참조하여 이름 충돌이 발생한 경우에도 완전한 질의 결과를 보내도록 한다.

▶ 관계 충돌(Relation conflict)

- 서브클래스충돌(subClassOf conflict)

동일한 클래스의 서브클래스 요소가 상이하여 충돌이 발생한 경우에는 충돌이 발생한 클래스들 간의 관계에 대한 판단이 요구된다. 충돌이 발생한 클래스간에 상하 관계가 존재한다면 통합 스키마를 구성할 때 계층구조에 의하여 해결한다. 즉, 스키마 구성시에 충돌 클래스간에 서브클래스 관계를 구성하여 줌으로서 자연스럽게 통합 스키마가 구성되도록 한다. 그러나, 두 개의 클래스간에 계층적 관계가 성립하지 않는다면, 다중상속을 허용하고, 시스템의 질의 처리시에 이를 고려하여 각 사이트별로 질의를 처리하도록 유도하는 방법이 있다.



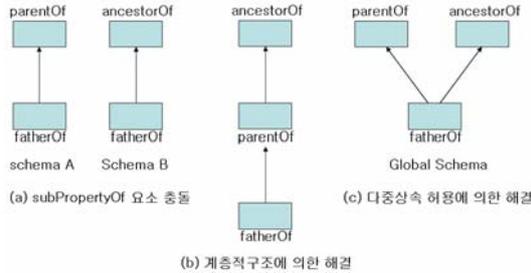
(그림 1) 서브클래스 충돌의 해결

(그림 1)은 서브클래스충돌의 해결 예를 보여준다. Taxi 클래스의 subClassOf 요소가 스키마에 따라 상이한 경우에, (b)에서 처럼 Vehicle과 Motorvehicle 간에 계층적인 의미가 존재한다면 두 개의 클래스간에 subClassOf 관계를 성립시켜 해결한다. 그러나, 이러한 관계가 성립하지 않으면 (c)와 같이 다중상속을 고려한 형태로 저장하여 처리하도록 제어한다.

- 서브프로퍼티충돌(SubpropertyOf conflict)

동일한 프로퍼티의 서브프로퍼티 요소가 상이하여 충돌이 발생한 경우에는 클래스 충돌과 유사하게 처리된다. 즉, 충돌이 발생한 프로퍼티들 간의 관계에 대한 판단이 요구된다. 충돌이 발생한 프로퍼티간에 상하 관계가 존재한다면 통합 스키마를 구성할 때 계층구조에 의하여 해결한다. 즉, 스키마 구성시에 충돌 프로퍼티간에 서브프로퍼티 관계를 구성하여 줌으로서 자연스럽게

계 통합 스키마가 구성되도록 한다. 그러나, 두 개의 프로퍼티간에 계층적 관계가 성립하지 않는다면, 다중상속을 허용하고, 시스템의 질의 처리시에 이를 고려하여 각 사이트별로 질의를 처리하도록 유도하는 방법이 있다.



(그림 2) 서브프로퍼티 충돌의 해결

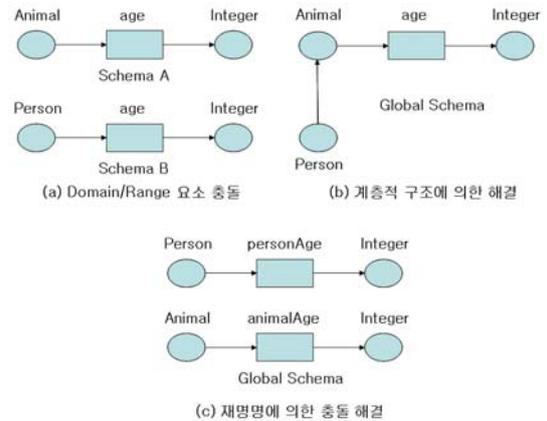
(그림 2)는 서브프로퍼티 충돌의 해결 예를 보여준다. fatherOf 프로퍼티의 subPropertyOf 요소가 스키마에 따라 상이한 경우에, (b)에서 처럼 parentOf와 ancestorOf 간에 계층적인 의미가 존재한다면 두 개의 프로퍼티간에 subPropertyOf 관계를 성립시켜 해결한다. 그러나, 이러한 관계가 성립하지 않으면 (c)와 같이 다중상속을 고려한 형태로 저장하여 처리하도록 제어한다.

- 도메인/레인지 충돌(Domain/Range conflict)

이름이 동일한 프로퍼티의 도메인이나 레인지 요소가 상이하여 충돌이 발생한 경우에는 충돌이 발생한 클래스간에 상하 관계가 존재한다면 통합 스키마를 구성할 때 계층구조에 의하여 해결 한다. 즉, 스키마 구성시에 충돌 클래스간에 서브클래스 관계를 구성하여 줌으로서 자연스럽게 통합 스키마가 구성되도록 한다. 만약에 이러한 관계가 성립하지 않는다면, 프로퍼티의 이름을 재명명하고 이를 중앙 카탈로그에 저장하여 질의 처리시에 참조하여 처리하도록 한다.

(그림 3)은 도메인/레인지 충돌의 해결 예를 보여준다. age 프로퍼티의 도메인 요소가 스키마에 따라 상이한 경우에, (b)에서 처럼 Animal과 Person 간에 계층적인 의미가 존재한다면 두 개의 클래스간에 subClassOf 관계를 성립시켜 해결한다. 그러나, 이러한 관계가 성립하지 않으면 (c)와 같이 personAge와 animalAge로 재명명하

여 저장하여 처리하도록 제어한다.



(그림 3) 도메인/레인지 충돌의 해결

▶ 타입 충돌(Type Conflict)

분산된 사이트의 스키마간에 타입충돌이 발생하면, 각 지역스키마의 이름을 파악하고 매칭시킬 수 있는 카탈로그 형태의 정보를 유지한다. 이때 각 사이트의 스키마내의 이름들을 구분할 수 있도록 충돌이 발생한 타입의 이름들을 각각 재명명하여 충돌을 제어한다. 카탈로그는 광역스키마에 대한 질의를 처리 하는 경우 반드시 참조하여 타입 충돌이 발생한 경우에도 완전한 질의 결과를 보내도록 한다.

```

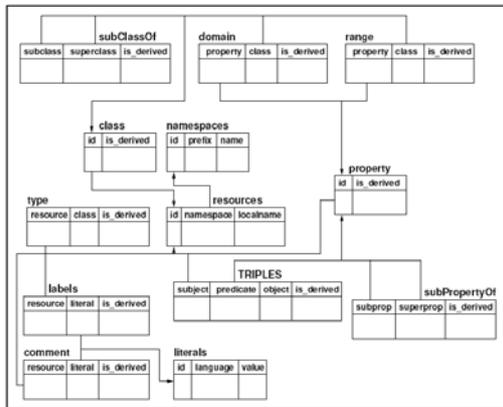
Schema A
<rdf:Description ID="Register">
<rdf:type resource=
"http://www.w3.org/2000/01/rdf-schema#Class"/>
<rdfs:subClassOf rdf:resource=
"#ComputerComponent"/>
</rdf:Description>
Schema B
<rdf:Description ID="register">
<rdf:type resource=
"http://www.w3.org/1999/02/22-rdf-syntax-ns
#Property"/>
<rdfs:domain rdf:resource="#Person"/>
<rdfs:range rdf:resource="#Product"/>
</rdf:Description>
Global Schema
<rdf:Description ID="ComputerRegister">
<rdf:type resource=
    
```

```

"http://www.w3.org/2000/01/rdf-schema#Class"/>
<rdf:subClassOf rdf:resource=
"#ComputerComponent"/>
</rdf:Description>
<rdf:Description ID="registering">
<rdf:type resource=
"http://www.w3.org/1999/02/22-rdf-syntax-ns
#Property"/>
<rdf:domain rdf:resource="#Person"/>
<rdf:range rdf:resource="#Product"/>
</rdf:Description>
    
```

2.4 스키마 저장구조

RDF로 표현된 메타데이터의 저장에 관한 연구는 기개발된 XML 데이터베이스 시스템에 RDF 데이터를 저장하는 연구와 상용 DBMS를 이용하여 RDF 데이터와 스키마를 저장하는 연구가 활발하게 진행되고 있다. 이 중에서 상용 DBMS를 이용한 방법은 RDF 스키마에 정의된 각각의 클래스와 프로퍼티 별로 독립적인 테이블을 생성하여 RDF 데이터의 정보를 저장하는 방법이다. On-To-Knowledge 프로젝트의 일부로 수행된 Sesame 시스템은 RDF로 표현된 메타데이터의 저장 및 질의 처리시스템이다[14]. (그림 4)는 관계형 데이터베이스 형태로 RDF/S를 저장하기 위한 테이블 구조이다.

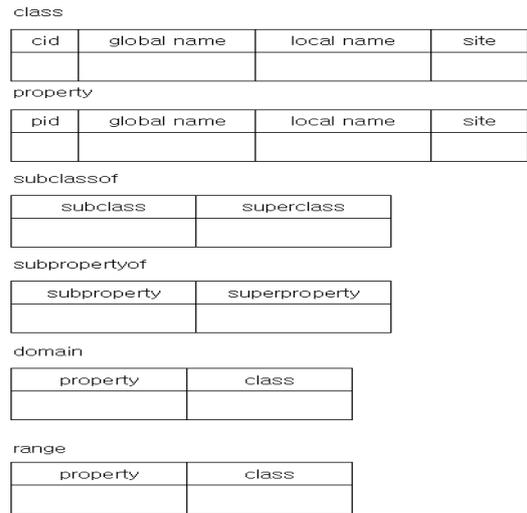


(그림 4) Sesame 시스템의 관계형 테이블 저장구조

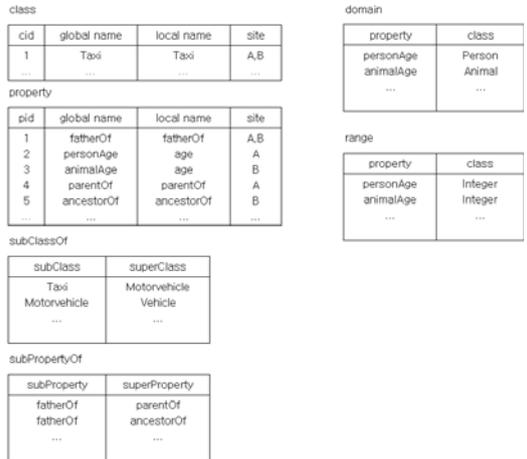
본 절에서는 Sesame와 같이 RDF/S를 관계 데이터베이스의 테이블로 표현된 여러 사이트의 데이터웨어하우스 환경을 충돌 제어가 가능한 하나의 통합된 메타데이터 스키마로 표현하기

위한 표현 구조를 제안하는 것이다. 앞 장에서 지적하고 제안된 여러 스키마의 통합에서 발생하는 여러 충돌 문제점들을 극복하기 위한 구조를 제안한다.

다음 (그림 5)는 스키마충돌 해결을 위한 데이터 저장 구조이다. class와 property 테이블에서 이름 충돌을 해결하기 위하여 로컬이름과 전역 이름을 구분하도록 구성하였다. 나머지 테이블에서는 앞장에서 제안된 선택적인 표현 방법으로 스키마를 표현하도록 구성하였다.



(그림 5) 충돌해결을 위한 테이블 저장구조



(그림 6) 충돌해결을 위한 테이블 저장 예

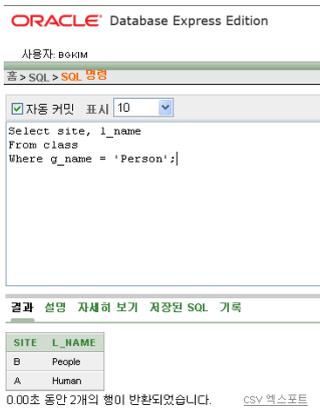
(그림 6)은 앞장에서 예를 들어 설명한 스키마들에 대한 제안된 표현 저장구조를 이용한 표

현 예이다.

2.5 통합스키마에 대한 질의

앞장에서 제시한 통합 스키마는 각 사이트간의 충돌을 제어하여 입력된 전역질의를 각 사이트에 안전하게 배분하고 통합하는 역할을 하기 위한 것이다. 본 장에서는 RDB의 테이블 형태로 표현된 통합 스키마에 대한 질의처리 예를 보여준다. 각 예는 앞장에서 설명한 충돌의 경우에 통합 전역스키마의 역할을 보여준다. 실험 환경은 CPU는 Intel Pentium D 3.40 Gz, RAM 1G, OS는 XP Professional, DBMS는 Oracle 10g Express 이다. 각 테이블에 입력된 실제 값들은 앞장의 예제 스키마를 표현한 값들이다. 다음 질의 예는 전역 스키마를 대상으로 하는 질의 내용을 각 사이트로 전송하여 각 지역 스키마를 대상으로 하는 질의 내용으로 변환하려고 할 때 필요한 질의의 처리 예를 보여준다.

- 클래스 이름 충돌 해결에 관련한 질의 : “Person” 개념이 존재하는 각 사이트와 각 사이트에서 지역 온톨로지에 어떤 이름으로 기술되어 있는지 검색하시오.



- 서브클래스 충돌 해결에 관련한 질의 : “Animal” 타입의 리소스들을 검색하기 위해서 질의를 전송해야하는 사이트를 검색하시오.



- 서브프로퍼티 충돌 해결에 관련한 질의 : “parentOf” 프로퍼티의 값들을 검색하기 위해서 질의를 전송해야하는 사이트를 검색하시오.



- 도메인/레인지 충돌 해결에 관련한 질의(계층적 구조에 의한 해결) : “age” 프로퍼티의 도메인 클래스 타입에 속하는 리소스 정보를 검색하기 위해서는 어떤 사이트에 대해 질의를 전송해야 하는가?



- 도메인/레인지 충돌 해결에 관련한 질의(재명명에 의한 해결) : “personAge” 프로퍼티의 도메인 클래스 타입에 속하는 리소스 정보를 검색

하기 위해서는 어떤 사이트에 대해 질의를 전송해야 하는가?



다음은 전역 스키마 내용 자체에 초점을 맞춘 질의의 예이다.

- 클래스 계층 구조에 대한 질의 : “Motorvehicle”의 상위 클래스를 검색하라.



- 도메인 관계에 대한 질의 : “Person” 클래스를 설명하기 위한 프로퍼티들을 검색하라.



3. 결론 및 향후 과제

본 연구에서는 분산된 환경에서 데이터웨어하우스를 구축하기 위하여 각 사이트별로 존재하는 지역 RDF 스키마를 통합할 때 발생하게 되는 스키마간의 충돌을 해결하는 방안을 제시하였다. 스키마 통합 시 발생할 수 있는 충돌의 종류를 이름충돌, 관계충돌, 타입충돌로 분류하였고 이에 대한 해결 방안을 제시하였다. 이름충돌의 경우에는 재명명 방법을 사용하였고, 관계 충돌의 경우에는 계층적인 관계를 파악하여 해결하거나, 다중상속을 허용하는 방식을 선택하도록 하였다. 타입충돌의 경우에는 재명명 방법을 사용하여 전역사이트에서 충돌을 제어하도록 하였다. 또한 사이트간의 스키마 충돌을 제어하기 위한 데이터 저장 구조를 제안하여 분산된 데이터웨어하우스의 통합 환경이 이루어지도록 하였다.

본 논문에서 분류하여 제시한 충돌은 RDF 스키마간에 발생할 수 있는 가장 일반적이고 빈번한 것들이며 이외에 세부적인 별도의 충돌들도 발생할 수 있다. 추가적인 연구를 통하여 실제로 발생 가능한 경우들을 추가할 것이다.

본 연구의 내용은 지역 온톨로지들을 통합한 전역 스키마를 구성하고 유지하는데 중요한 역할을 할 것이다. 더 나아가 전역 스키마를 통하지 않고 지역 스키마들간의 통신을 통하여 질의를 처리하고자 할 때에도 적용이 가능하도록 추가적인 연구가 요구된다.

참고 문헌

- [1] Genesereth M, Keller A, and Duschka O, "Informaster: An information integration system", ACM SIGMOD, 1997.
- [2] Nimble Technology, "The Nimble Integration Suite", White Paper, 2002.
- [3] Metamatrix, "Model-Driven Information Integration", White Paper, 2002.
- [4] Vincent Aguilera, Sophie Cluet, and Fanny Watzet, "Xyleme Query Architecture", WWW Posters, 2001.
- [5] W3C, "Resource Description Framework(RDF) Model and Syntax Specification", (Available at <http://www.w3.org/TR/1999/REC-rdf-syntax-19990222/>), 1999.

[6] W3C, "A Model-Theoretic Semantics for DAML+OIL", (Available at <http://www.w3.org/TR/2001/NOTE-daml+oil-model-20011218>), 2001.

[7] W3C, "OWL Web Ontology Language Overview", (Available at <http://www.w3.org/TR/2004/REC-owl-features-20040210/>), 2004.

[8] Sudha Ram, Jinsoo Park, "Semantic conflict resolution ontology (SCROL): an ontology for detecting and resolving data and schema-level semantic conflicts", IEEE Transactions on Knowledge and Data Engineering, Volume 16, Issue 2, pp 189 - 202, Feb. 2004

[9] 박현미, 박석, "XML SSchema를 이용한 스키마 통합에서 충돌문제의 포괄적 분류와 해결", 정보과학회 가을 학술발표논문집, pp 184-186, 2002.10.

[10] Jeongsoo Lee, Heekwon Chae, Kwangsoo Kim, and Cheol-Han Kim, "An Ontology Architecture for Integration of Ontologies", LNCS 4185, ASWC 2006, pp. 205-211, 2006

[11] Dou D, Macdermot D, Qi P, "Ontology Translation on the Semantic Web", LNCS Journal of Data Semantics, Vol. 2, No. 3360, pp. 35-57, 2004.

[12] Ehrig, M., Sure Y., "Ontology Mapping - An Integrated", ESWS 2004, pp 76-91, 2004.

[13] N. Noy and M. Musen, "The PROMPT suit: interactive tools for ontology merging and mapping", International Journal of Human-Computer Studies, vol. 59, pp 983-1024, 2003.

[14] Jeen Broekstra, Arjohn Kampman, and Frank van Harmelen, "Sesame: A Generic Architecture for Storing and Querying RDF and RDF Schema", International Semantic Web Conference, pp 54-68, 2002.

김 병 곤



1990년 : 홍익대학교 공과대학
전자계산학과 이학사
1992년 : 홍익대학교 공과대학
전자계산학과 대학원 이학석사
2001년 : 홍익대학교 공과대학
전자계산학과 대학원 이학박사

1992년~1998년 : 국방과학연구소 연구원
2001년~현재 : 부천대학 e-비즈니스과 조교수
관심분야 : 다차원 인텍싱, 데이터웨어하우스, 시맨틱 웹 등