XML Schema기반 시맨틱 데이타 통합
(An XML Schema-based Semantic Data Integration)

김 동 광 † 정 갑 주 †† 신 효 섭 ††† 황 선 태 ††††
(Dongkwang Kim) (Karpjoo Jeong) (Hyoseop Shin) (Suntae Hwang)

요 약 과학·공학 분야의 사이버 인프라스트럭쳐는 다양한 도메인에서 수행되는 연구 활동을 통해서 얻어지는 다양한 형식의 데이타들뿐만 아니라 이런 데이타를 저장·관리하기 위한 이질적인 저장소들의 통합이 요구되고 있다. 데이타 통합 작업의 어려움은 다음과 같다: (1) 시스템 독립적인 다중 데이타 스키마 지원, (2) 다양하게 변화하는 스키마들의 쉬운 관리, (3) 직관적인 스키마 맵핑. 이 같은 문제를 해결하기 위해서, 우리는 XML Schema를 이용해서 과학 분야의 데이타 모델을 정의하고 RDF기반의 스키마 맵핑을 이용해서 의미적으로 통합할 수 있는 새로운 방법을 제안한다. XML Schema기반의 데이타 모델 정의 방법은 실험 데이타들을 과학자들이 직관적이고 간편하게 표현 할 수 있게 해주며, 이 데이타 모델은 많은 시스템에서 사용중인 XML DBMS를 그대로 이용할 수 있는 장점이 있다. 또한, 스키마 맵핑을 위해서 RDF로 구축된 온톨로지를 이용해서 XML Schema로 정의되어 있는 스키마의 구조적인 관계를 정의하고, 맵핑 정보를 이용해서 통합 질의를 수행한다. 우리는 제안 시스템의 프로토타입을 토목 공학 분야 프로젝트인 KOCED에 적용하였다.

키워드 : 과학 기술 데이타 관리, 데이타 통합, 시맨틱 통합, XML 스키마, 그리드

Abstract Cyber-infrastructures for scientific and engineering applications require integrating heterogeneous legacy data in different formats and from various domains. Such data integration raises challenging issues: (1) Support for multiple independently-managed schemas, (2) Ease of schema evolution, and (3) Simple schema mappings. In order to address these issues, we propose a novel approach to semantic integration of scientific data which uses XML schemas and RDF-based schema mappings. In this approach, XML schema al-lows scientists to manage data models intuitively and to use commodity XML DBMS tools. A simple RDF-based ontological representation scheme is used for only structural relations among independently-managed XML schemas from different institutes or domains We present the design and implementation of a prototype system developed for the national cyber-environments for civil engi-neering research activities in Korea (similar to the NEES project in USA) which is called KOCEDgrid (http://www.koced.net).

Key words : Scientific Data Management, Data Integration, Semantic Integration, XML Schema, Grid

1. 서 론

과학·공학 분야의 실험 과정은 다양하며 상호 독립적인 태스크들로 구성된 수행 시간이 길고 복잡한 워크플로우로(workflow)이다. 각각의 태스크들은 수행되면서 실험과 관련된 정보(예, 실험 파라미터, 설정 정보 등)들과 다양한 종류의 실험 결과 데이타(예, 센서 정보, 이미지, 영상 정보 등)들을 생성한다. 그래서 과학 데이타는 실험을 통해서 얻어지는 결과 데이타뿐만 아니라, 실험을 수행하면서 요구되고 생성되는 실험 정보들까지 포함한 것을 의미한다. 과학 기술 데이타의 데이타 모델은 실험 구성 요소들의 형태를 정의하고 이들을 구조적으로 정리하고 논리적으로 표현해 놓은 것을 의미하며, 이런 데이타 모델을 이용해서 실험 절차에 대한 연관성을 효율적으로 표현할 수 있어야 된다. 스키마는 데이타 모델을 시스템에 표현해 놓은 것을 의미한다. 예를 들면, 우리가 의미하는 데이타 모델은 관계형 DBMS에서

† 학생회원 : 건국대학교 컴퓨터공학과
walhalla@gcslab.konkuk.ac.kr
†† 종신회원 : 건국대학교 인터넷멀티미디어 공학부 교수
jeongk@konkuk.ac.kr
††† 정 회 원 : 건국대학교 인터넷미디어공학부 교수
hsshin@konkuk.ac.kr
†††† 종신회원 : 국민대학교 컴퓨터학부 교수
sthwang@kookmin.ac.kr
논문접수 : 2006년 5월 24일
심사완료 : 2006년 8월 17일

의 DDL과 같다.

e-Science분야에서의 과학 데이타 통합은 다른 조직들과 지역들간의 데이타 통합을 의미하는 것으로 최신의 기술들이 필요하고 어려운 일이다. 이 같은 통합은 과학자들에게 다른 시스템의 스키마에 상관없이 다양한 데이타 저장소의 데이타를 일관적인 방법으로 접근할 수 있는 것을 의미하는 것으로, 이 같은 데이타 통합은 현재까지 연구되어오고 있는 전통적인 데이타 관리 방법으로는 매우 어려운 일이다[20].

과학 기술 데이타의 데이타 모델구축과 통합을 위해서는 다음과 같은 문제들이 해결되어야 한다:

1) 다양하게 변화하는 스키마들의 쉬운 관리: 과학 분야에서는 새로운 연구 주제들이 끊임없이 생겨나며 연구를 통해서 좋은 결과를 얻기 위해서 과학자들은 다양한 방식의 새로운 실험 과정을 적용한다. 그러므로, 이런 다양한 형태의 연구 과정을 통해서 얻어지는 데이타들을 관리하기 위해서 데이타 모델(스키마)은 계속적으로 수정되게 된다. 이 같은 스키마의 변화에 대해서 과학자들이 새로운 것과 기존 스키마의 데이타들을 효과적으로 관리 할 수 있는 방법이 필요하다. 그러나 전통적인 관계형 DBMS에서는 스키마 변화를 관리하기가 어렵다.
2) 시스템 독립적인 다중 데이타 스키마 지원: 다양한 연구 기관들간의 통합 인프라스트럭쳐인 e-Science와 그리드 시스템에서 데이타를 통합 관리하기 위해서 중앙 집중적인 방법으로 스키마를 관리하는 것은 매우 어려운 일이다. 그러므로, 시스템들간 독립적으로 스키마들을 정의하고 관리할 수 있는 분산된 방식의 스키마 관리방법이 필요하다.
3) 직관적인 스키마 맵핑: 과학 분야에서의 데이타 통합은 과학자들이 다른 형태의 스키마로 표현되어 있는 데이타들을 통일된 방법으로 검색 할 수 있도록 해야 된다. 이를 위해서 각 스키마들의 요소들간에 연관성을 정의해 주어야 한다(이것을 스키마 맵핑이라고 한다).

데이타 통합을 지원하기 위해서 최근 들어 많은 분야에서 연구가 이루어지고 있다. BIRN Project는 뇌에 대한 다양한 분야(인간의 뇌에 관련된 질병과 형태와의 관계에 대한 연구, 정신분열증에 대한 이미지 분석, 질병에 걸린 쥐의 뇌 연구)의 연구 데이타들을 통합하기 위한 프로젝트이다[1]. 또한, GEONgrid는 렌더링된 지도 데이타를 통합 검색하기 위해서 공간적인 데이타들에 대해서 온톨로리를 구축하고 이것을 이용해서 데이타 질의를 수행할 수 있는 정보 처리 상호 운영을 위한 시스템을 개발하였다[18].

이 논문에서는, 이질적인 데이타 자원으로부터 다양한 타입의 데이타들을 그리드 시스템간에서 효율적으로 검색하고 공유하기 위해서 XML Schema를 이용해서 멀티 데이타 모델을 정의하는 방법을 제안하고, 데이타 모델의 구조적인 차이점으로 인해서 통합 검색할 수 없는 XML Schema기반의 데이타 모델을 RDF/RDF Schema를 이용해서 구조적 차이점을 표현하는 통합 온톨로지를 구축하고, 이를 기반으로 해서 통합 검색하는 방안을 제안한다.

2. 방법론

2.1 XML스키마 기반의 데이타 모델

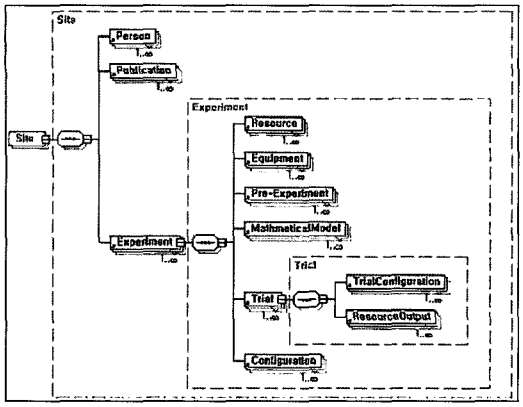
데이타 모델링을 설계할 때 가장 많이 사용되는 방법으로는 Entity-Relationship(E-R) Model, Object-Oriented Model, XML based Model등이 있다[4-6]. 이 논문에서는 최근에 많은 과학 분야에서 데이타와 메타 데이타를 표현하고 정의하기 위해서 사용되는 XML Schema를 이용해서 데이타 모델을 표현할 것이다. 또한, 일반적인 XML Schema기반의 데이타 모델링 기법과 더불어 데이타 모델의 구조적 관계를 정의하기 위해서 RDF(Resource Description Framework)를 사용할 것이다.

그러나 우리는 데이타 모델을 표현하기 위해서 XML Schema를 이용하고, RDF는 오직 스키마 맵핑 하는 데에만 이용할 것이다. 왜냐하면 데이타 모델을 RDF로 표현하는 것보다 XML Schema를 이용하는 것이 좀더 직관적이며, RDF의 추론을 처리하기 위한 오버헤드를 최소화 할 수 있기 때문이다. 또한 XML Schema기반의 데이타 모델은 XML 문서를 관리하기 위해서 많이 사용되는 XML DBMS[33]를 이용할 수 있으며, XQuery와 같은 질의 처리 언어를 이용해서 빠르게 데이타를 검색할 수 있다[9,10].

그림 1은 XML Schema로 정의되어 있는 데이타 모델을 보여준다. 이 데이타 모델은 NEES project[3]에 의해서 개발된 것으로 현재 KOCED project[2]에서도 사용되고 있다. NEES와 KOCED는 토목 공학 실험을 원격 모니터링하고 데이타를 관리하기 위해서 중앙의 데이타 저장소를 이용해서 통합하려는 프로젝트이다. 이 데이타 모델은 실험 과정의 일반적인 형태와, 토목 공학의 실험 정보를 표현하고 있다. 그러나 여기서 표현되지 않은 다른 많은 종류의 데이타들이 연구를 진행하면서 생성되거나, 설정 정보로 만들어 질것이고, 이 데이타 모델은 이런 변화에 유연하게 대처할 수 있다.

2.2 스키마 관리

우리가 제안하는 방법에서는 다양한 실험 시설의 스키마를 독립적으로 관리하는 것을 가정한다. 이 스키마 관리 방법은 그림 2에서 자세히 볼 수 있다. 과학자가




```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qua
  <xs:element name="Site" type="Site"/>
  <xs:complexType name="Site">
    <xs:sequence>
      <xs:element ref="Person" maxOccurs="unbounded"/>
      <xs:element ref="Publication" maxOccurs="unbounded"/>
      <xs:element ref="Experiment" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Experiment" type="Experiment"/>
  <xs:complexType name="Experiment">
    <xs:sequence>
      <xs:element ref="Resource" maxOccurs="unbounded"/>
      <xs:element ref="Equipment" maxOccurs="unbounded"/>
      <xs:element ref="Pre-Experiment" maxOccurs="unbounded"/>
      <xs:element ref="MathmeticalModel" maxOccurs="unbounded"/>
      <xs:element ref="Trial" maxOccurs="unbounded"/>
      <xs:element ref="Configuration" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Trial" type="Trial"/>
  <xs:complexType name="Trial">
    <xs:sequence>
      <xs:element ref="TrialConfiguration" maxOccurs="unbounded"/>
      <xs:element ref="ResourceOutput" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="Person"/>
```

그림 1 KOCED project의 데이타 모델

많은 스키마에 대한 자세한 정보 없이도 간단하게 질의를 입력(Global Schema Management)할 수 있어야 되고 전체 시스템에 대해서 통합 질의를 수행(Query Generation)하는 것이 필요하다. 중앙의 Global Schema Management방식의 시스템 디자인은 사용자들이 각 시스템에 존재하는 다중 스키마에 대한 자세한 정보 없이도 질의를 생성할 수 있도록 정보를 제공해준다.

앞에서 설명했던 것처럼, 과학 데이타들을 관리하기 위해서 스키마의 변화(이것은 곧 실험 과정의 변화이다)하기 때문에 스키마 관리는 어려운 일이다. 데이타 베이스 분야에서는 이미 스키마의 변화에 대해서 오래 전부터 연구가 진행되어왔으나, 현재까지도 전통적인 데이타베이스 시스템에서는 해결하기 어려운 문제로 남아있다[30]. 그래서 우리가 제안하는 방법은 각 시스템 별로 스키마를 독립적으로 관리한다. 예를 들어서 실험 사이트 A에서 스키마를 생성했다고 가정하면, 사이트 B에서는 이 스키마를 이용해서 새로운 스키마를 생성할 수 있다. 우리 시스템은 이 두 개의 스키마를 독립적으로 관리할 뿐만 아니라, 각 스키마의 데이타를 검색 위해서 통합 질의를 생성하고 이 질의어를 각 사이트의 스키마에 맞는 질의로 변환하여 검색할 수 있도록 지원한다. 그림 2는 이 같은 시스템 디자인을 설명하고 있으며, 그림 3은 XML 스키마의 변화에 대해서 설명하고 있다.

다중 스키마(Multiple Schema)를 통합 검색 할 수 있도록 통합 질의를 생성하기 위해서, 우리는 스키마 맵핑 기술을 이용한다. 우리는 다중 스키마들간의 구조적 차이점을 설명하기 위한 온톨로지를 구축하기 위해서 XML 스키마를 RDF로 표현해서 관리할 수 있도록 시스템을 디자인 하였다. 우리가 제안하는 방법과 기존 RDF기반의 온톨로지 방법과의 차이점은 우리는 온톨로지를 구축하기 위한 기본정보인 개념(Concept)과 관계(Relationship)정보를 표현하지 않고, 계층 구조적의 스키마에 정의되어 있는 요소(Element)들간의 간단한 구조적 차이점을 표현함으로써 스키마 맵핑을 한다.

3. 시스템 디자인

3.1 시스템 모델

그림 4는 우리가 제안하는 시스템의 전체 과정을 설명하고 있다. 우리가 제안하는 시스템은 두 단계로 이루

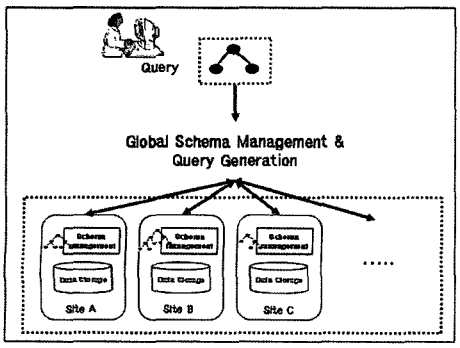


그림 2 글로벌 스키마 관리 방법(좌)

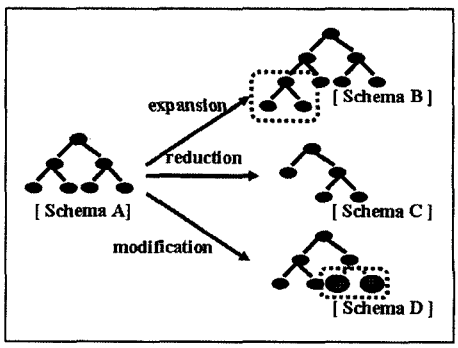


그림 3 스키마의 변화(우)

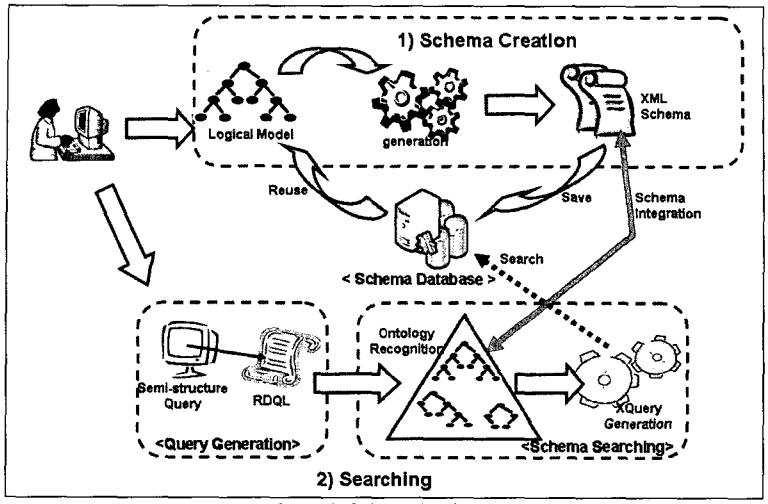


그림 4 데이타 통합 시나리오

어져 있다. 과학 데이타를 저장하기 위해서 Metadata의 데이타 모델을 정의하고 등록한 후에 통합 Ontology를 구축하는 작업과, 구축된 통합 Ontology를 이용해서 입력되어 있는 Metadata를 의미적(Semantic)으로 검색하는 작업으로 구성되어 있다.

첫 번째 단계는 과학 데이타(실험의 기본적인 정보와 실험 결과 데이타)들을 저장하기 위해서 Metadata의 데이타모델을 정의하고, Schema Database에 등록하고, 기존에 등록되어있는 데이타 모델들과 새로운 데이타 모델간의 구조적 차이를 기술하는 작업이다. 좀더 자세히 설명하면, 먼저 과학자들은 수행하려는 실험에 대해서 실험 시설, 실험장비, 참여자, 실험 횟수, 실험 조건 등을 포함하는 Metadata의 데이타 모델을 정의하고, 이를 XML Schema 기반의 데이타 모델로 작성한다. 이렇게 작성된 Metadata모델은 시스템에 등록된다. 새롭게 등록된 데이타 모델은 시스템에 등록되어 있는 기존 데이타 모델과 구조적 차이점을 비교해서 서로 동일하거나 차이가 있는 Element들에 대해서 통합 온톨로지를 구축하게 된다. 이렇게 작성된 Metadata schema와 온톨로지 정보는 Schema Database(XML Database)에 저장된다. 또한 새로운 실험을 수행하려는 과학자가 시스템에 등록되어 있는 실험의 Metadata schema에서 자신이 수행하려는 실험과 비슷하거나 연관성을 지니고 있는 schema가 정의되어 있는지 검색한다. 검색된 Schema를 기반으로 자신의 실험에 맞춰 수정해서 새로운 Metadata schema를 생성하고 등록할 수 도 있다; 그림 4의 1) Schema Creation.

두 번째는, 구축된 온톨로지 정보를 이용해서 통합 검색 과정이다. 앞에서 설명한 것과 같이 다양한 실험이 수행하면서 새로운 스키마가 생성되거나, 기존 스키마가 수정되면서 스키마의 구조적 차이를 표현하기 위해서 통합 Ontology가 구축되고 저장, 관리된다. 사용자는 이렇게 구축된 Ontology정보를 이용해서 데이타를 검색한다. 사용자는 자신이 찾고자 하는 검색 항목과 키워드를 입력하면 시스템은 이에 맞는 RDQL(Resource Description Query Language)[13]을 생성되게 된다. 이렇게 생성된 RDQL을 실행해서 구축되어 있는 Ontology의 구조적 관계를 분석하고, 상관 관계를 찾아내어 XML Schema의 구조적인 정보를 파악한 다음에, 해당 Schema에 맞는 XQuery를 생성하게 된다. 결과적으로 생성된 XQuery를 실행해서 얻어진 Metadata들을 취합해서 사용자에게 보여지게 된다; 그림 4 2) Searching.

3.2 스키마 맵핑

XML Schema기반의 데이타 모델은 다른 데이타 모델간의 구조적인 차이점으로 인해서 통합하는 것이 어렵기 때문에, 이를 극복하기 위해서 우리의 스키마 맵핑 기법은 XML Schema로 정의되어 있는 데이타 모델의 요소들을 의미적인 관계로 표현함으로써 구조적 충돌을 해결하였다. 우리는 계층적 구조의 데이타 모델의 요소들간 구조적 관계를 정의해서 통합 온톨로지를 구축하였다. 좀더 자세히 설명하면, 데이타 모델의 요소들의 구조적인 관계를 RDF[15]로 표현해서 통합 온톨로지를 구축하는 것이다. 다른 데이타 모델의 모든 요소들의 구조적 차이는 다음의 3가지 관계로 정의 될 수 있다.

1. 이름은 다르지만 의미가 같은 것: Schema A에서 정의한 Element와 Schema B에서 정의되어 있는 Element가 정의된 이름은 다르지만, 의미하는 것은 같은 경우 (/A/B equal /A'/B').
2. 이름과 의미가 같은 경우: Schema A의 element와 Schema B의 Element가 구조적으로 같고, 이름과

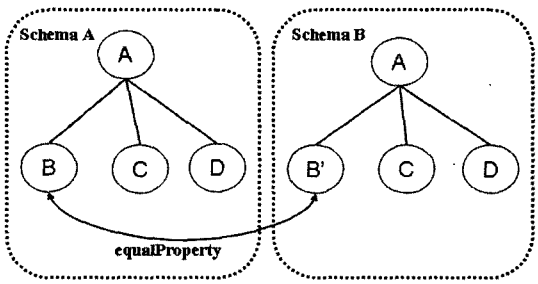


그림 5 다른 스키마들간의 구조적 차이점

의미가 같은 경우 (두 Schema의 /A/D는 같다)

우리는 이런 관계를 표현하기 위해서 RDF를 사용하였다. RDF는 온톨로지를 구축하기 위한 표현 언어중의 한 종류로서, 모든 요소를 Triples(subject, predicate, object)의 관계로 정의하여 표현한다. 예를 들어 책의 이름이 'Semantic Web'이고, 저자가 'Tim Berners-Lee'인 책이 있다고 할 경우에, 책이 subject가 되고, 'Semantic Web'과 'Tim Berners-Lee'가 object로, subject와 object를 연결하는 관계인 이름과 저자가 predicate로 정의된다.

Metadata의 element들을 RDF로 표현하기 위해서 우리는 아래와 같은 변환 규칙을 정의 했다. 데이타 모델의 모든 요소들을 subject로 정의하고 이들의 하위 요소(sub-element)들을 object로 정의하였다. 그리고 이들간을 predicate를 이용해서 관계를 설정하였다. 우리는 앞에서 언급한 각 요소들간의 연관성과 element와 attribute들 간의 관계를 정의하기 위해서 3개의 predicate로 정의하였다.

1. hasProperty: subject인 Element와 object인 Element에 정의되어 있는 Attribute들간의 관계를 나타내는 predicate
2. equalProperty: 두 개의 Metadata사이에서 이름은 다르지만 의미가 같은 Element들의 관계를 나타내는 predicate

그림 6에서는 위와 같이 정의한 predicate를 이용해서 XML Schema로 정의된 Metadata의 Element들을 RDF로 변환하여 표현하고, 이들간의 관계를 설정한 예제를 볼 수 있다.

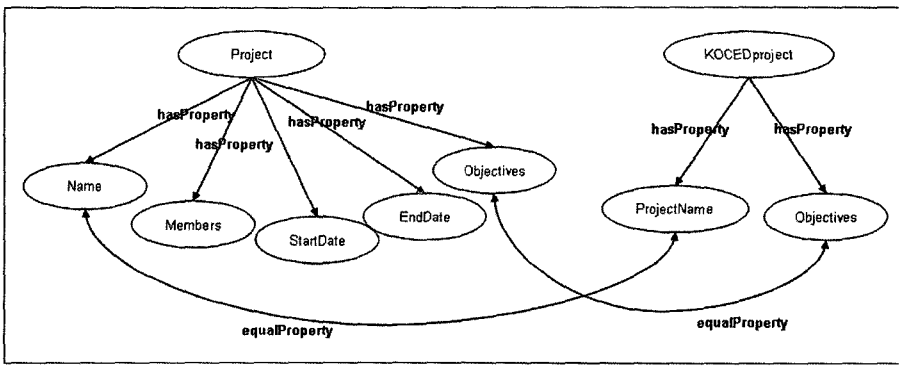


그림 6(a) RDF 그래프 예제

```
.....
<rdf:Description rdf:about="http://gcslab.konkuk.ac.kr/2006/kincholaw/1.0#Project">
  <semantic:hasProperty rdf:resource="http://gcslab.konkuk.ac.kr/2006/kincholaw/1.0#Project/Name"/>
  <semantic:hasProperty rdf:resource="http://gcslab.konkuk.ac.kr/2006/kincholaw/1.0#Project/Members"/>
  <semantic:hasProperty rdf:resource="http://gcslab.konkuk.ac.kr/2006/kincholaw/1.0#Project/StartDate"/>
  <semantic:hasProperty rdf:resource="http://gcslab.konkuk.ac.kr/2006/kincholaw/1.0#Project/EndDate"/>
  <semantic:hasProperty rdf:resource="http://gcslab.konkuk.ac.kr/2006/kincholaw/1.0#Project/Objectives"/>
</rdf:Description>
<rdf:Description rdf:about="http://gcslab.konkuk.ac.kr/2006/kincholaw/1.0#KOCEDProject">
  <semantic:hasProperty rdf:resource="http://gcslab.konkuk.ac.kr/2006/kincholaw/1.0#KOCEDProject/ProjectName"/>
  <semantic:hasProperty rdf:resource="http://gcslab.konkuk.ac.kr/2006/kincholaw/1.0#KOCEDProject/Objectives"/>
</rdf:Description>
<rdf:Description rdf:about="http://gcslab.konkuk.ac.kr/2006/kincholaw/1.0#Project/Name">
  <semantic:equalProperty rdf:resource="http://gcslab.konkuk.ac.kr/2006/kincholaw/1.0#KOCEDProject/ProjectName"/>
</rdf:Description>
<rdf:Description rdf:about="http://gcslab.konkuk.ac.kr/2006/kincholaw/1.0#KOCEDProject/ProjectName">
  <semantic:equalProperty rdf:resource="http://gcslab.konkuk.ac.kr/2006/kincholaw/1.0#Project/Name"/>
</rdf:Description>

.....
```

그림 6(b) RDF로 표현한 예제

우리가 제안하는 방법은 심플한 스키마 맵핑 기법이다. 새로운 Metadata가 추가되거나, 기존의 Metadata가 수정돼서 등록될 경우에, schema의 모든 element를 RDF로 변화해서 통합 Ontology에 추가하는 것은 아니다. 추가되는 schema에 대해서는 기존 schema와의 구조적인 차이점만을 기술한다. 데이타 통합을 정의하는 방법에는 대표적으로 2개의 방법이 있다. 첫 번째 방법은 Global Schema를 필요로 하는 GAV(global-as-view)로서, 이것은 모든 데이타의 관계를 정의하고 있다. 더욱이, Global Schema의 모든 Element들은 데이타 소스와 직접적으로 관련성을 표현함으로써, 데이타를 검색할 때 데이타가 저장되어 있는 데이타 소스를 직접적으로 검색할 수 있게 해주고 있다. 두 번째 방법은 각 데이타들에 대한 스키마를 정의하고 이들과는 독립적으로 Global Schema를 구축하는 LAV(local-as-view)이다. Global Schema와 데이타 소스와의 관계는 Global Schema상에서 각각의 데이타 소스의 조건들을 기술함으로써 구축된다. 그러나 이 같은 방법은 약점이 있다. GAV는 스키마의 크기가 커지게 되고, 계속적으로 변화하는 스키마와 같은 경우에는 적합하지 않으며, LAV는 Global Schema와 데이타소스와의 연관성을 구축하는 것이 어렵다. 반면에 우리가 제안한 방법의 경우에는 XML Schema로 데이타 모델을 구축하고, 이들간의 구조적 차이점만을 RDF로 기술 함으로서 Global Schema를 구축할 필요가 없을 뿐만 아니라, 이렇게 구축된 통합 온톨로지 정보를 이용해서 통합 검색을 할 수 있다.

3.3 시맨틱 검색

의미적(Semantic) 통합 검색은 데이타 모델간의 구조적 차이에 관계없이 전체 스키마에 대한 검색이 가능한 것을 의미한다. 다양한 XML Schema기반의 데이타 모델은 구조적 차이점으로 인해서 XQuery와 같은 구조적 질의어를 이용해서는 통합 검색을 할 수 없다. 그래서 데이타 모델들 간의 구조적 관계를 정의하고 이들간의 관계를 통합 온톨로지로 표현한다. 우리는 이러한 통합 온톨로지를 이용해서 데이타베이스에 저장되어 있는 데이타를 시맨틱하게 검색할 수 있다.

시맨틱 검색은 두 단계로 이루어져 있다. 첫 번째 단계는 사용자가 검색하고자 하는 질의를 입력한다. 입력된 질의어는 RDQL 언어로 변환되어 스키마 데이타 베이스에서 실행되고 구축되어 있는 온톨로지의 요소와 스키마의 구조적인 관계를 분석하게 된다. 두 번째 단계는 RDQL의 실행 결과로 얻어진 정보들을 이용해서실제 데이타 소스를 검색하기 위한 XQuery언어가 생성된다. RDQL은 RDF 언어로 구축되어 있는 온톨로지를 검색할 수 있는 언어이다. 이것은 SQL언어와 비슷하고 검색하고자 하는 요소들을 관계정보인 triple(subject, predicate, object)로 표현하여 검색할 수 있는 기능을 제공한다. 우리는 구조적 관계를 표현하고 있는 통합 온톨로지를 구축하고 이들을 XQuery와 같은 구조적 검색언어가 아니라 RDQL언어를 이용해서 검색함으로써 같은 의미이면서 다른 구조적 특징을 가지고 있는 데이타 모델을 통합 검색 할 수 있도록 하였다(그림 7).

Inserted query
element1 = 'search word #1' (AND|OR)
element2 = 'search word #2' (AND|OR)
element3 = 'search word #3'

```
while not end(insertQuerys) {
  // Step.1
  equalResult = select ?x where ( insertQuerys[i], equalPropery , ?x );
  // Step.2
  whlie not end(equalResult) {
    equalStructure = select ?y where ( ?y , hasProperty, equalResult[j] );
  }
  // Step.3
  generated_xquerys[i] = RDFtoXQuery( equalStructure );
}
// Step.4
execute generated_xquerys[];
```

그림 7 입력 질의어와 온톨로지 검색 알고리즘

Semantic Searching의 처리 과정은 다음과 같다. 생성된 RDQL은 입력된 질의의 요소(Element)들 간의 구조적 관계를 검색하도록 되어있다. 이렇게 생성된 RDQL을 순차적으로 실행해서 equalProperty관계가 있는 데이타 모델의 Element들을 찾아낸다(Step.1). 검색된Element들은 자신과 predicate가 hasProperty관계인 subject를 찾아서 구조적인 관계를 파악하고(Step.2), 이런 관계를 분석해서 XQuery를 생성한다(Step.3). Ontology로 구축된 Element들은 각자 자신이 존재하는 Schema에 대한 정보들을 유지하고 있기 때문에, 해당 Element에 맞는 XQuery가 만들어지는 것이다. 이렇게 생성된 XQuery들은 해당 Schema가 저장되어 있는 데이타 베이스에 실행되고(Step.4), 검색된 Metadata들은 통합되어 사용자에게 보여지게 된다.

4. 스키마 맵핑 예제

이번 장에서는 실험 연구과정에서 새로운 스키마가 시스템에 등록되고, 어떻게 두 개의 스키마의 구조적 정보가 온톨로지로 표현되는지간략한 예를 들어서 설명할 것이다. 'Metadata schema A'에 'Equipments', 'Sensor', 'DAQBoard'인 총 3개의 Element가 정의되어 있다. 'Equipments'는 ComplexType(XML Schema로 정의한 새로운 Type의 자료형)이며 'Sensor'와 'DAQBoard'를 sub-element로 소유하고 있는 Parent객체이고, 'Sensor'는 xsd:string(XML Schema의 기본 데이타 타입)으로 정의되어 있고 이 Element가 의미하는 것

은 Sensor의 이름이다(그림 8(a)). 실험을 수행하려는 다른 사용자는 시스템에 등록되어 있는 'Schema A'의 'Sensor'에 대한 정의가 수행하려는 실험의 스키마와 일치 하지 않아서 새로운 Schema B를 정의하였다. 'Sensor'를 Complex Type으로 정의하고, 'Name'과 'Values'라는 값을 추가하였다. 'Name'과 'Values'는 각각 기본 데이타 타입인 xsd:string, xsd:float으로 정의해서 'Sensor' Element에 추가하였다(그림 8(b)). 두 개의 XML Schema에 정의되어 있는 'Equipments'와 'DAQBoard'는 구조와 요소의 의미가 같고, Schema A의 'Sensor'와 Schema B의 'Sensor/Name'은 구조적으로는 차이점가 있으나, 의미하는 것은 같다. 그러나 이런 구조적인 차이점으로 인해서 사용자가 XQuery와 같은 검색어를 이용해서 'Sensor = X310a'와 같은 데이타를 찾고자 할 경우에는 두 스키마의 구조적 차이점으로 인해서 Schema A에 대해서만 검색된 결과를 얻을 수 있게 된다.

이러한 구조적인 차이점을 극복하고 전체 스키마에 대해서 통합 검색하기 위해서는 두 스키마의 Element중 의미적으로 같으나 구조적으로 다른 Element들에 대한 구조적 관계 정보를 통합 온톨로지로정의 해야 된다. 먼저 Schema A에 대해서 RDF문장을 이용해서 subject가 'Equipments'이고, 'Sensort(xsd:string)'와 'DAQBoard'가 object로 정의되어있고, 이들간의 predicate는 'hasProperty'로 정의되었다. 이렇게 구축된 Ontology는 Metadata B가 등록되면서 다음과 같이 수정되어야 한다. 새롭게 추가되는 ComplexType인 'Sensor(type:sensor)'를 'Equipments'에 추가해야 된다. 'Sensor(type:sensor)'를 object로 정의하고 'Equipments'와 hasProperty로 predicate를 연결한다. 또한 'Sensor(type:sensor)'가 가지고 있는 Sub-Elemenet인 'Name(xsd:string)'과 'Values(xsd:float)'를 Ontology로 표현하기 위해서, 'Sensor(xsd:sensor)'를 subject로, 'Name(xsd:string)'과 'Values(xsd:float)'를 object로 하고, hasProperty를 predicate로 정의해서 새로운 Description을 추가한다. 새롭게 추가된 'Sensor(type:sensor)'의 'Name(xsd:string)'과 기존에 존재하는 'Sensor(xsd:string)'가 서로 같은 의미를 가지고 있기 때문에 두 Element에 대해서 앞에서 설명했던predicate중에 'equalProperty'로 관계를 설정하게 된다. 그림 9의 오른쪽 그림이 완성된 통합 Ontology이다.

사용자는 데이타를 검색하기 위해서 완성된 통합 Ontology를 이용하게 된다. 좀더 구체적으로 Sensor의 이름이 'X310a'인 Metadata를 검색하고자 한다면, 사용자가 입력하는 질의어는 다음과 같다.

'Sensor = X310a'

이렇게 입력된 질의어는 먼저 구축된 Ontology의 Element인 'Sensor(xsd:string)'와 equalProperty의 관계로 연결되어 있는 Element를 검색하기 위한 RDQL로 변경된다. 생성된 RDQL의 실행 결과로 predicate가 equalProperty인 'Name(xsd:string)'을 찾아내게 된다. 그리고 나서 'Name(xsd:string)'과 hasProperty관계인 'Sensor(type:sensor)'를 검색하게 되고 결국 Sensor/Name이라는 계층적 구조를 파악하게 된다. 이렇게 파악된 구조 정보를 이용해서 XQuery가 생성되게 된다.

```
for $x in doc('instances1.xml')/InstanceData
where contains( $x//Equipments/Sensor, 'X310a')
return $x

for $x in doc('instances2.xml')/InstanceData
where contains ($x//Sensor/Name, 'X310a')
return $x
```

이 XQuery는 각 Metadata가 저장되어 있는 데이타베이스에 실행되고, 실행 된 결과는 통합되어 사용자에게 보여진다.

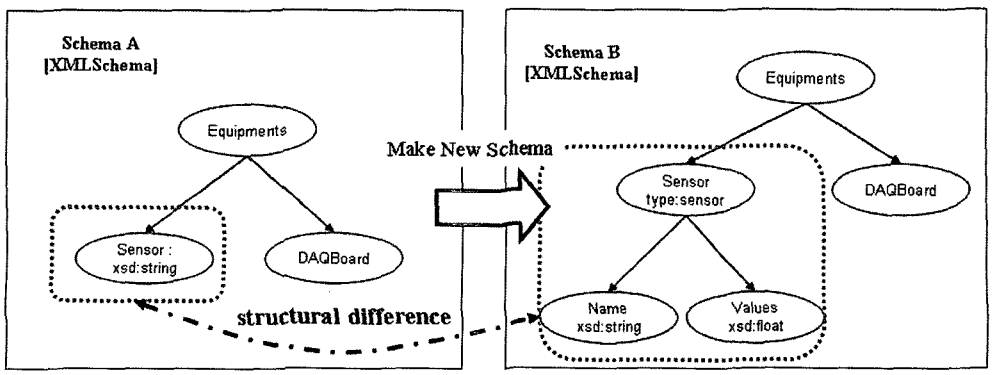


그림 8 (a) 스키마 A, (b) 스키마 B

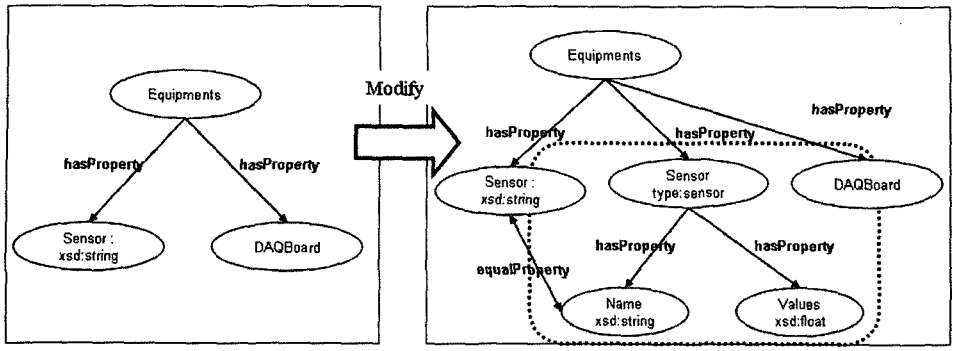


그림 9 통합 Ontology의 변화

5. 구 현

이번 장에서는 우리가 제안하는 시스템의 아키텍처와 시맨틱 데이타 관리를 위해서 구축된 인터페이스에 대해서 설명한다.

5.1 시스템 아키텍처

그림 10은 우리가 제안하는 시맨틱 데이타 통합의 시스템 아키텍처를 설명하고 있다. 우리의 시스템은 크게 2개의 component로 구성된다.

다양한 데이타 모델을 관리하기 위한 Schema Management와 통합 온톨로지를 기반으로 한 검색 시스템이다. Schema Management는 Schema Register/Viewer/Modifier와 Metadata 입력 시스템, Ontology Management로 구성된다. 각각은 데이타 모델을 생성하기 위한 Schema Registry와 구축되어 있는 데이타 모델을 관리하고 편집하기 위한 Schema Modifier, 관리자들이 데이타 모델을 관리하기에 편의성을 제공하기 위해서 XML Schema를 GUI로 표시해주는 Schema Viewer이다. 또한 구축되는 데이타 모델에 대해서 통합 온톨로지로 구축하기 위한 Ontology Management가 있다. 이렇게 정의되는 데이타 모델과 Ontology정보는 OGSA-DAI[16]를 통해서 각 시스템에서 사용하는 데이타 베이스에 저장되고 관리될 수 있다. OGSA-DAI는 이질적인 데이타 베이스 환경에 대해서 통합 질의 할 수 있는 미들웨어 시스템이다.

통합 검색은 RDF based-Query Generator와 Ontology & Schema Searching component에 의해서 이루어 진다. 먼저 RDF base-Query Generator는 사용자가 통합 검색을 할 수 있도록 통합 온톨로지를 분석하여 제공하고, 사용자가 선택한 온톨로지 정보를 기반으로 해서 전체 데이타 모델의 부분 구조 정보를 이용해 검색 질의어인 RDQL을 생성하게 된다. 여기서 생성된 질의어들은 Ontology&Schema Searching component로 전달되어, 이를 데이타 모델이 저장되어 있는 각 데이타 베이스에 맞는 XQuery의 질의어로 변환되게 된다. 최

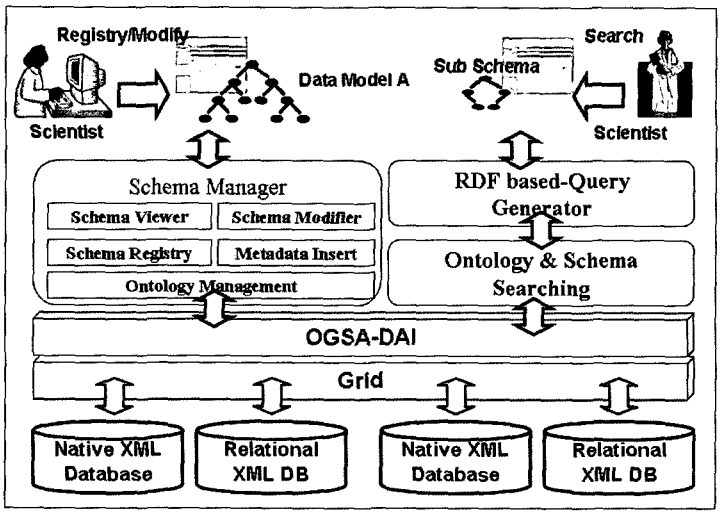


그림 10 시맨틱 데이타 통합 아키텍처

종적으로 생성된 XQuery는 OGSA-DAI를 통해서 각 데이터 베이스에서 실행되고, 수행된 결과는 취합되어 사용자들에게 보여지게 된다.

5.2 웹 기반의 시맨틱 데이터 관리 사용자 인터페이스

우리는 건축토목분야의 그리드 시스템인 KOCED[8]에 본 논문의 방법을 적용하였다. KOCED의 다양한 건축·토목의 연구 분야에서 실험 정보를 공유하기 위해 각 실험에 맞는 데이타 모델을 정의하고 관리하기 위한 XML Schema기반의 Tool을 구현하였다. 이 Tool을 이용해서 새로운 데이타 모델을 XML Schema문법에 대해서 알고 있지 않아도 쉽게 작성할 수 있고, 또한 기존에 등록된 Schema를 검색해서 자신이 표현하고자 하는 데이타 모델로 수정하거나, 새로운 데이타 모델로 등록할 수 있게 구현하였다. 시스템에 등록되어 있는 메타데이타의 데이타 모델을 가지고 실제 데이타를 입력할 수 있도록, 데이타 모델 기반의 입력 폼을 자동으로 생성하고, 입력할 수 있는 기능을 제공하며, 이렇게 입력된 데이타는 XML Database에 저장되고 관리된다. 또한 새로운 데이타 모델을 추가할 때, 현재까지 구축된 통합 Ontology에 Mapping하기 위한 Tool을 제공하고 있다. 이를 이용해서 사용자는 간편하게 새롭게 추가된 Element와 기존의 통합 Ontology와 Mapping작업을 처리할 수 있다.

또한, 사용자들이 데이타 모델에 대한 정보 없이 시스템에 등록되어 있는 메타데이타를 검색할 수 있도록, 구축된 Ontology를 검색 조건으로 보여주고, 이 조건 중에 하나를 선택해서 질의어를 작성 할 수 있다. 그리고 하나의 조건만이 아닌 복수의 질의어를 입력하여, 효율적인 검색이 가능하도록 구현하였다. 이렇게 생성된 검색 질의어는 OGSA-DAI[9]를 통해서 그리드 시스템의 전체 데이타 베이스에 통합 질의를 실행하고 결과를 취합해서 사용자에게 보여준다.

6. 관련 연구

NEES와 GEONgrid는 연구 결과 데이타 관리와 이들 데이타를 이용한 협업하기 위한 그리드 시스템 이다[17,18]. NEES(The Network for Earthquake Engineering Simulation)는 참조 데이타 모델을 정의해서 이 데이타 모델을 기반으로 분산되어 있는 기관들간의 데이타를 관리하고 공유하기 위해서 NEESCentral을 구축하였다. 이 NEESCentral은 실험 시설로부터의 연습 및 실험 결과 데이타들 기반의 일반적인 데이타 모델을 이용해서 데이타를 관리 할 수 있는 웹 기반의 서비스를 제공한다. 또한, 이 시스템은 계층적 모델을 제공함으로써, 개별적으로 데이타들을 디렉터리기반으로 관리할 수 있는 기능을 제공하고 있다. 그러나 NEESCentral은 실험에 대한 전체적인 데이타 모델을 제시하고 있지 않기 때문에 과학자들이 데이타를 효과적으로 관리 할 수 없으며, 데이타 모델이 고정되어 있기 때문에 과학자들의 실험을 수행하면서 새롭게 표현되는 데이타 모델을 적용할 수 없다. GEONgrid(Geosciences Network)[3]는 지구과학 분야의 공동 연구를 위한 이미지 데이타를 통합 관리하기 위한 그리드 시스템으로서, 이

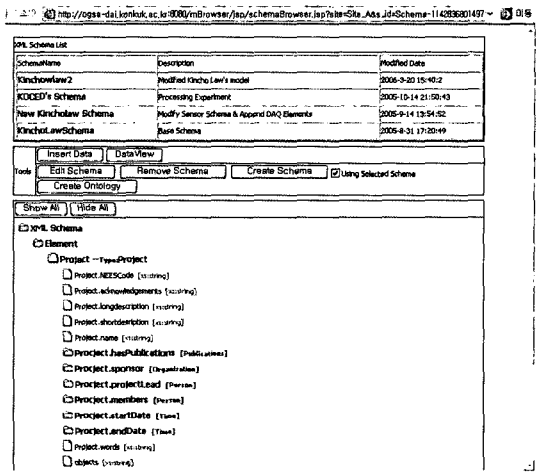


그림 11(a) 스키마 브라우져 UI

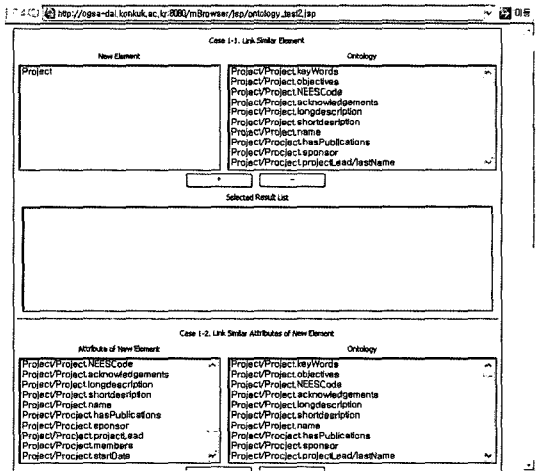


그림 11(b) 온톨로지 맵핑 UI

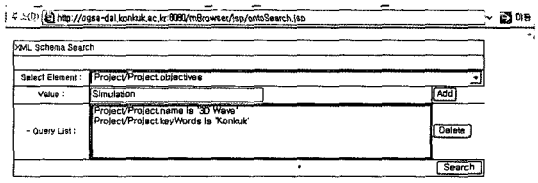


그림 11(c) 메타데이타 검색 UI

지도 정보인 이미지들을 통합 관리하고, 일관적인 사용자 인터페이스를 제공하여 통합 검색할 수 있도록 구축하였다. 이 시스템은 이런 데이타들을 관리하기 위해서 OWL을 이용해서 온톨로지 기반의 검색을 구현하였다. 이 온톨로지는 지질학적인 특징들로 정리되어 있으며, 새롭게 추가되는 지도 이미지들을 구축된 분류 정보인 온톨로지와 연결시킴으로써, 사용자들이 통합 검색을 할 수 있기 하였다. 그러나 이 프로젝트는 데이타를 통합한 것이 아니라, 지구 과학의 분류 기준으로 사용되는 정보들을 온톨로지로 구축하고, 연구를 통해서 얻어지는 이미지 데이타들과 온톨로지로 표현된 분류 정보들의 관계를 온톨로지로 표현해서 관리하고, 분류 정보를 검색 조건으로 이용한 검색이 가능하도록 구축되어 있다.

EDUTELLA와 Piazza는 다양한 포맷의 데이타들을 사용하게 되는 다른 시스템들간의 정보 상호 운용 서비스를 제공하는 시스템들이다[19, 20]. EDUTELLA는 P2P Network상에서 의미적으로 데이터를 공유를 목표로 하는 시스템이다. 이 시스템은 RDF를 이용해서 실제 데이터를 정의하고, RDQL(RDF질의어)을 이용해 데이터를 검색한다. 이기종 시스템의 데이터들 검색하기 위해서 정의되어 있는 변환 모듈을 통해서 입력된 RDQL질의들은 해당 데이터를 검색할 수 있는 형태로 번역되고 최종적으로 검색을 수행하는 방식이다. 그러나 EDUTELLA는 이기종 시스템간의 데이터를 맵핑을 위한 방법을 미리 정의해야 되기 때문에 광범위하거나 변화가 자주 발생하는 데이터들의 통합 검색하기에는 한계를 드러낸다. Piazza는 XML로 표현되어 있는 데이타들을 공유하기 위해서 XML간의 구조적인 차이점을 설명하는 XQuery와 유사한 언어를 개발하였다. 이 언어를 이용해서 각 XML문서의 차이점을 기술하고, 자동 번역 시스템을 통해서 각 시스템간의 데이타를 공유하도록 구축되어 있다.

7. 결론 및 향후 계획

그리드 시스템에서의 데이타 통합은 과학 분야의 효율적인 공동연구를 위해서 꼭 해결 되야 하는 중요한 부분이다. 다양한 형태의 데이타들을 쉽게 표현하고, 연구 기관들간에 효율적으로 접근하기 위해서는 통합 검색이 필요하다.

이 논문에서는 이질적인 데이타에 대한 효율적 공유를 위한 XML Schema기반의 데이타 통합 방법을 제안하였다. 먼저 같은 연구 분야의 데이타들이 실험 시설이나 실험 조건에 따라서 다양하게 정의되야하는 특징이 있는 Metadata를 확장성(flexible)있는 XML Schema를 이용해 정의하였다. 또한, 사용자가 다양한 데이타 모델을 통합 검색하기 위해서 시스템에 등록되는 Metadata 데이타 모델의 각 Element간의 관계만을 통합 Ontology로 구축함으로서, 계속 변화하는 Schema에 대한 효율적인 관리와, 검색 방안을 개발하였다.

우리가 제안한 방안인 XML Schema로 정의되어 있는 데이타 모델과 RDF로 표현되는 Ontology와의 Mapping과 통합 검색 시에 선택하는 Ontology에 대해서, 사용자가 쉽게 접근 할 수 있는Visual Tool의 개발에 대해서 계속적인 연구가 필요하다.

참 고 문 헌

[1] Mark Ellisman and Steve Peltier: Medical Data Federation: The Biomedical Informatics Research Networks, The Grid 2 Second Edition, Pages: 109-120, 2004.

[2] Korea Construction Engineering Development Collaboration(KOCED), www.koced.net

[3] Jun Peng, Kincho H. Law: Reference NEESgrid Data Model [TR-2004-40] (2004).

[4] Prentice-Hall, NJ : Database Design Using Entities and Relationships, Chen, P. P., S. B. Yao (ed.), Principles of Data Base Design, 1985, pp. 174-210.

[5] J. Arlow and I. Neustadt. UML and the Unified Process: Practical Object-Oriented Analysis and Design, Addison-Wesley Pub Co., Boston, MA, 2001.

[6] Tim Bray and C.M. Sperberg-McQueen, "Extensible Markup Language (XML): Part I. Syntax," World Wide Web Consortium Recommendations, February 1998, Available at http://www.w3.org/TR/REC-xml.

[7] M. Fernandez, W.-C. Tan, and D. Suciu. Silk-Route: Trading between relations and XML. In Ninth International World Wide Web Conference, November 1999.

[8] David C. Fallside, "XML Schema Part 0: Primer," World Wide Web Consortium Candidate Recommendation, October 2000, Available at http://www.w3.org/TR/xmlschema-0/.

[9] S.Boag, D.Chamberlin, M.F.Fernandez : XQuery 1.0: An XML query Language, 30 Aprial 2002, http://www.w3.org/TR/xquery

[10] Z. G. Ives, A. Y. Halevy, and D. S. Weld. An XML query engine for network-bound data. VLDB Journal, 11(4):380-402, December 2002.

[11] T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web. Scientific American, May 2001.

[12] P. Patel-Schneider and J. Simeon. Building the Semantic Web on XML. In Int'l Semantic Web Conference '02, June 2002.

[13] Andy Seaborne, HP Labs Bristol : RDQL - A Query Language for RDF, 9 January 2004, http://www.w3.org/Submission/RDQL/

[14] A. Doan, J. Madhavan, P. Domingos, and A. Halevy. Learning to map between ontologies on the semantic web. In Eleventh International World Wide Web Conference, 2002.
[15] Dan Brickley and R. V. Guha.: W3C Resource Description Framework(RDF) Schema Specification, http://www.w3.org/TR/1998/WD-rdf-schema/, March 2000. W3C Candidate Recommendation.
[16] Mario Antonioletti, Malcolm Atkinson, Rob Baxter, Andrew Borley: The design and implementation of Grid database services in OGSA-DAI(Database Access and Integration Services), Concurrency and Computation: Practice & Experience archive Volume 17, Issue 2-4, Pages: 357-376 (2005).
[17] NEESgrid, http://it.nees.org
[18] GEONgrid, http://www.geongrid.org
[19] Wolfgang Nejdl, Boris Wolf, Changtao Qu : EDUTELLA: A P2P Networking Infrastructure Based on RDF (2002), May 7-11, 2002, WWW 2002.
[20] Zachary G. Ives, Alon Y. Halevy, Peter Mork : Piazza: Mediation and Integration Infrastructure for Semantic Web Data, Journal of Web Semantics manuscript.
[21] Phd thesis, University of Bremen. in German : Semantic Mediation for heterogeneous Information Sources. (2003).
[22] J. Broekstra, A. Kampan, and F. van Harmelen. Sesame : A generic architecture for storing and querying RDF and RDF Schema., In Int'l Semantic Web Conference '02, pages 54~68, 2002.
[23] T. R. Gruber. A translation approach to portable ontology specifications. Knowledge Acquisition, 5:199-220, 1993.
[24] B. Amann, C. Beeri, I. Fundulaki, and M. Scholl. Ontology-based integration of XML web resources. In Int'l Semantic Web Conference '02, pages 117-131, 2002.
[25] E. Mena, V. Kashyap, A. P. Sheth, and A. Illarramendi. OBSERVER: An approach for query processing in global information systems based on interoperation across pre-existing ontologies. Distributed and Parallel Databases, 8(2):223-271, 2000.
[26] A. Y. Levy, A. Rajaraman, and J. J. Ordille. Querying heterogeneous information sources using source descriptions. In VLDB '96, pages 251-262, 1996.
[27] A. P. Sheth and J. A. Larson. Federated database systems for managing distributed, heterogeneous, and autonomous databases. ACM Computing Surveys, 22(3):183-236, 1990.
[28] D. D. Roure, I. Foster, E. Miller, J. Hendler, and C. Goble. The semantic grid: The grid meets the semantic web. Panel at the WWWConference, Honolulu, Hawaii, 2002.
[29] Yannis Kalfoglou and Marco Schorlemmer: Ontology mapping: the state of the art, The Knowledge Engineering Review, Volume 18, Issue 1, Pages: 1-31 (2003).
[30] Avi Silberschatz, Henry F. Korth, S. Sudarshan: Database System Concepts Fifth Edition, ISBN 0-07-295886-3
[31] Bachler, M., Buckingham-Shum, S., Chen-Burger, J., Dalton, J., Roure, D. D., Eisenstadt, M., Frey, J., Komzak, J., Michaelides, D., Page, K., Potter, S., Shadbolt, N. and Tate, A., Chain ReAKTing: Collaborative Advanced Knowledge Technologies in the CombeChem Grid. in UK e-Science All Hands Meeting, (Nottingham, UK, 2004).
[32] Hughes, G., Mills, H., de Roure, D., Frey, J., Moreau, L., schraefel, m. c., Smith, G. and Zaluska, E: The semantic smart laboratory: a system for supporting the chemical eScientist. Organic and Biomolecular Chemistry 2:pp. 1-10. (2004).
[33] Bellahsene Zohra, Milo Tova, Rys Michael, Suciu Dan, Unland Rainer: Database And XML Technologies , Second International Xml Database Symposium, Xsym 2004, Toronto, Canada, August 29-30, 2004, Proceedings.

김 동 광
2005년 2월 건국대학교 컴퓨터 공학과 (학사). 2005년 3월~현재 건국대학교 컴퓨터 공학과(석사과정). 관심분야는 Grid Computing, Semantic Data Integration

정 갑 주
정보과학회 논문지 : 시스템 및 이론
제 33 권 제 7 호 참조

신 효 섭
1994년 2월 서울대학교 컴퓨터공학과 학사. 1996년 2월 서울대학교 컴퓨터공학과 석사. 2002년 2월 서울대학교 전기컴퓨터공학부 박사. 1999년 3월~2000년 2월, 2001년 1월~2001년 5월 미국 University of Arizona 전산학과 방문연구원. 2001년 7월~2005년 2월 삼성전자 소프트웨어센타 책임연구원. 2005년 3월~현재 건국대학교 인터넷미디어공학부 조교수. 관심분야는 Semantic Web, Scientific Data Management, XML Database, Multimedia Database, TV Anytime

황 선 태
정보과학회 논문지 : 시스템 및 이론
제 33 권 제 7 호 참조